(xcommodore

## VIC-1001 USER'S MANUAL

**PERSONAL COMPUTER** 

by commodore

# PERSONAL COMPUTER USER'S MANUAL

(x commodore

#### ●注意●

- (1)本書の無断転載は禁止いたします。
- (2)内容は予告なく変更することがあります。
- (3)本書の内容は充分にチェックしておりますが、万一誤り、記載もれ、御不審な点など、お気付きのことがありましたら御連絡下さい。
- (4)(3)項にかかわらず、使用した結果の影響については、責任を負いかねますので御了承下さい。

#### はじめに

本書には、VIC-1000シリーズ・パーソナル・コンピューター・システムを初めて使用される方のために、コンピューターの基本的なしくみ、使用方法、ハードウェアの構成、VIC-1001パーソナル・コンピューターのプログラミング言語 CBM BASIC (VERSION 2) 等に関する説明がなされています。

各種のオプションを付加することにより、あらゆる目的に用いることのできる VIC-1960シリーズ・パーソナル・コンピューター・システムは、TVゲーム、パズル、個人的情報管理から各種科学技術計算、各種ビジネスデータの処理、シミュレーション等々まで、幅広い対応が可能となります。

#### はじめに

| <b>第1部</b> ハードウェア編                             | 9  |
|--|----|
| 第1章 コンピューターのしくみ                                | 10 |
| 第2章 VIC-1001 パーソナル・コンピューター                     |    |
| 第1節 VIC-1999シリーズ・システム構成                        |    |
| 第2節 VIC-19999シリーズ・ファミリー                        | 14 |
| 第 3 章 VIC-1gg!の使用方法······                      |    |
| 第1節 電源投入の前に                                    |    |
| 第2節 電源の投入                                      |    |
| 第3節 ディスプレイ                                     |    |
| 1. 画面の文字数                                      |    |
| 2. カラー   |    |
| 3. キャラクター・コード                                  |    |
| 第4節 キーボード・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・   |    |
| 1. キーボード配列・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・ |    |
| 2. キャラクター・モードの選択                               |    |
| 3. 特殊キーの働き                                     | 30 |
| 第4章 BASICによるプログラミング                            | 32 |
| 第1節 BASICとは                                    | 32 |
| 第2節 プログラムの書き方                                  | 32 |
| (1)ダイレクト・モード                                   | 33 |
| (2)プログラム・モード                                   | 35 |
| 第3節 プログラムの修正                                   | 37 |
| 第4節 スクリーン・エディター                                | 41 |
| 第5章 カセット・ドライブの使い方                              | 43 |
| 第1節 プログラムのSAVE                                 | 43 |
| 第2節 プログラムのVERIFY                               |    |
| 第3節 プログラムのL()A1)                               | 45 |
| * 4 8 = 7 OI OAD to ETSCAVE                    | 46 |

| 第6章 ハー     | - ドウェアの説明 51                     |
|------------|----------------------------------|
| 第1節        | 概論51                             |
| 第2節        | 外觀                               |
| 第3節        | システム構成 52                        |
| 第4節        | 機能任様54                           |
| 第5節        | メモリー・アドレスマップ 56                  |
| 第6節        | キーボード                            |
| 第7節        | インターフェイス63                       |
| 第8節        | CRT制御方式······ 67                 |
|            |                                  |
| 第2部 :      | ノフトウェア編69                        |
|            |                                  |
|            | M BASIC の概要70                    |
| 第1節        | 初期状態70                           |
| 第2節        | 動作モード・・・・・・・・・・・・・・・・・・・・・・・・・70 |
| 第3節        | ラインの書式71                         |
| 第4節        | ライン・ナンバー·······71                |
| 第5節        | キャラクター・セット71                     |
| 第6節        | 定数74                             |
|            | 牧形式74                            |
|            | <b>助小数点形式74</b>                  |
|            | 変数75                             |
| 1 1 1 1    | 数名および型宣言文字75                     |
|            | 河変数75                            |
| 第8節        | 型の変換76                           |
| 第9節        | 式と演算77                           |
|            | <b> </b>                         |
|            | 系演算子78                           |
|            | 里演算子78                           |
|            | 发                                |
|            | 字列の演算 80<br>スクリーン・エディター 81       |
|            |                                  |
| A2 1 7 161 | エラー・メッセージ・・・・・・・・・・・・・・・・・・81    |
|            |                                  |
| r          |                                  |

| 第2章 CB | M BASICのコマンドとステートメント82 |
|--------|------------------------|
| 2.1    | C L O S E 83           |
| 2.2    | C L R 84               |
| 2.3    | C M D 84               |
| 2.4    | C O N T 84             |
| 2, 5   | D A T A 85             |
| 2.6    | DEF FN86               |
| 2.7    | D ! M 87               |
| 2.8    | E N D 87               |
| 2.9    | FOR~NEXT               |
| 2.10   | GET, GET # 90          |
| 2.11   | GOSUB~RETURN91         |
| 2.12   | GOTO92                 |
| 2.13   | IF~THEN, IF~GOTO92     |
| 2.14   | I N P U T 94           |
| 2.15   | INPUT #95              |
| 2.16   | L E T 96               |
| 2 . 17 | L   S T 96             |
| 2.18   | L O A D 97             |
| 2.19   | NEW98                  |
| 2.20   | ON~GOSUB, ON~GOTO98    |
| 2.21   | O P E N 99             |
| 2.22   | P O K E100             |
| 2.23   | PRINT, PRINT #100      |
| 2.24   | R E A D102             |
| 2.25   | R E M104               |
| 2.26   | RESTORE104             |
| 2.27   | R U N104               |
| 2.28   | S A V E105             |
| 2.29   | S T O P105             |
| 2.30   | S Y S106               |
| 2.31   | VERIFYt06              |
| 2.32   | W A T T107             |

| 第3章  | CBN   | M BASICの関数108       |
|------|-------|---------------------|
| 3.   | 1     | A B S110            |
| 3.   | 2     | A S C 110           |
| 3.   | 3     | A T N110            |
| 3.   | 4     | C H R \$111         |
| 3.   | 5     | C O S111            |
| 3.   | 6     | E X P 111           |
| 3.   | 7     | F R E112            |
| 3.   | 8     | I N T 112           |
| 3.   | 9     | LEFT\$112           |
| 3.   | 10    | L E N               |
| 3.   | 11    | L O G113            |
| 3.   | 12    | M I D \$113         |
| 3.   | 13    | P E E K             |
| 3.   | 14    | P O S114            |
|      | 15    | R I G H T \$ 114    |
| 3.   |       | R N D115            |
|      | 17    | S G N115            |
| 3.   | 18    | S I N116            |
| 3.   | 19    | S P C116            |
| -    | 20    | S Q R116            |
| 3,   | 21    | S T A T U S117      |
| 3.   | 22    | S T R \$118         |
| 3.   | 23    | T A B118            |
| 3.   | 24    | T A N119            |
| 3.   | 25    | T I M E119          |
| 3.   | 26    | T I M E \$120       |
| 3.   | 27    | U S R               |
| 3.   | 28    | V A L121            |
|      |       |                     |
| 第4章  | ÇBI   | M BASICのプログラム例122   |
|      |       | りまして123             |
|      |       | SQUIGGLE (スクィグル)124 |
|      |       | - ディングその 1126       |
| 4. L | 7 7 - | - ディングその 2128       |

| 5   | , VICオルガン······129                   |
|-----|--------------------------------------|
| 6   | . 壁こわしゲーム131                         |
| 7   | . ひらがな132                            |
| 8   | . 文字パターン作成プログラム133                   |
|     |                                      |
| 付録  |                                      |
| Λ   | : CBM BASICのコード表136                  |
| В   | : CBM BASICの省略形······137             |
| C   | :他のBASICからCBM BASICへの変換138           |
| D   | :エラーメッセージ一覧表139                      |
| E   | :誘導関数147                             |
| F   | : カラーコントロール148                       |
| G   | :サウンド153                             |
| H   | : ハイ・レゾリューション155                     |
|     |                                      |
| 第 3 | 部 コンポーネント・データカタログ編163                |
|     |                                      |
| 1   | . MPS6500マイクロブロセッサー165               |
| 2   | . MPS656Øビデオ・インターフェイス・チップ (VIC) …181 |
| 3   | . MPS6522汎用インターフェイス・アダプタ199          |
| 4   | . MPS2364スタティックROM231                |
|     |                                      |

### 第1部 ● ハードウェア編

#### 第1章 コンピューターのしくみ

コンピューターを初めて使用される方々のために、そのしくみを簡単に説明しておきます。

コンピューターは、記憶装置と人・出力装置から構成されています。VIC-1001 の記憶装置を構成している主要部品は次の通りです。

CPU (CENTRAL PROCESSING UNIT) またはMPU (MICROPROCES SOR UNIT: MPS6502A×1

ROM (READ ONLY MEMORY): MPS2364×2および2332×1

RAM (RANDOM ACCESS MEMORY) : MPS2114×10

VIA (VERSATILE INTERFACE ADAPTOR) : MPS6522×2

VIC (VIDEO INTERFACE CHIP) : MPS6560×1

CPU (またはMPU) は、人間にたとえれば脳に相当する部分であり、上記の部品のコントロールをおこなっています。

ROMは、一定の動作をするためだけの部品です。BASICは、ROMに書き込まれており、たとえば、RUNと人力された時には、RAMの中にあるプログラムを実行する指示をCPUに与えます。電源を切っても、内容は消えません。

RAMは、ユーザー・プログラムおよびデータが格納される部分です。電源を切れば、内容は消えてしまいます。

VIAは、入・出力装置の制御をする部品です。

VICは、その名の示す通り、ビデオ・インターフェイス・チップであり、CRTへの表示をコントロールしています。

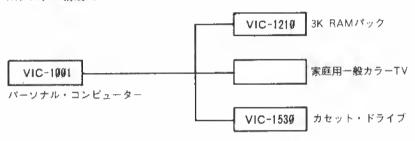
これらを中心とする記憶装置全体を人間の脳とすれば、目・耳に相当する部分を入・出力装置と呼び、VIC-16001本体に組み込まれているものに、キーボードがあります。キーボードは、人力のみのために使用されます。VIC-16001本体の外部につけられるの入・出力装置としてはCRTディスプレイ(家庭用一般カラーTVまたは専用カラー・モニター)、カセット・ドライブ、プリンター、ジョイスティック、ライトペン等があります。

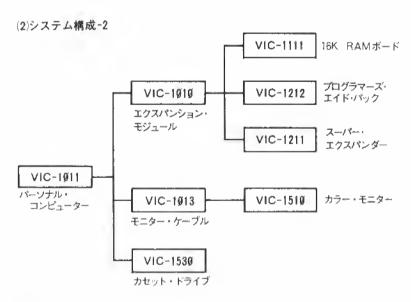
#### 第2章 VIC-1001パーソナル・コンピューター

#### 第1節 VIC-1000シリーズ・システム構成

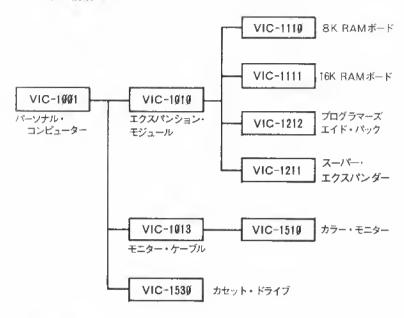
VIC-1000シリーズ・パーソナル・コンピューター・システムは、その用途に応じて、拡張可能です。以下では、その組合わせ例を図で説明します。

#### (1)システム構成-1

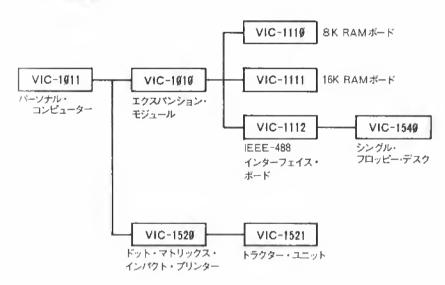




#### (3)システム構成-3



#### (4)システム構成-4



#### 第2節 VIC-1000シリーズ・ファミリー

VIC-1000シリーズ・パーソナル・コンピューター・システムには、以下のファミリーがあります。

| MODEL NO. | DESCRIPTION            |
|-----------|------------------------|
| VIC-1001  | パーソナル・コンピューター          |
| VIC-1010  | エクスパンション・モジュール         |
| VIC-1011  | RS-232C・アダプター・ボード      |
| VIC-1Ø12  | マルティプル・コントロール・ボード      |
| VIC-1013  | モニター・ケーブル              |
| VIC-1110  | 8K RAM・ボード             |
| VIC-1111  | 16K RAM・ボード            |
| VIO-1112  | IEEE-488インターフェイス・ボード   |
| VIC-1210  | 3K RAM・バック             |
| VIC-1211  | スーパー・エクスパンダー           |
| VIC-1211M | スーパー・エクスパンダー(3K RAM付き) |
| VIC-1213  | プログラマーズ・エイド・パック        |
| VIC-1310  | ライト・ペン                 |
| VIC-1311  | ジョイ・スティック              |
| VIC-1312  | パドル                    |
| VIC-1510  | カラー・モニター               |
| VIO-1520  | ドット・マトリックス・インパクト・プリンター |
| VIC-1521  | トラクター・ユニット             |
| VIC-153Ø  | カセット・ドライブ              |
| VIC-1540  | シングル・フロッピー・ディスク        |

● VIC-1010: エクスパンション・モジュール

エッジカード・コネクター 4 本実装 (2 本追加可能)。VIC-1991の接続ケーブル・ 電源・簡易キャビネットつき。VIC-1991のエキスパンション・バスを利用。

● VIC-1011:RS-232C・アダブター・ボード

RS-232Cを持つ各種機器と、VIC-1991を接続。VIC-1991のユーザーポートに接続。

●VIC-10/12:マルティブル・コントロール・ボード

VIC-1001のコントロール・ポートと接続。ジョイ・スティク4、接続可能。

● VIC-1Ø13:モニター・ケーブル

VIC-1601のカラービデオ・インターフェイスと専用カラーモニター接続用。家庭

用TV使用の場合は、RFモデュレーターについているケーブルを使用。

● VIC-111#: 8K RAMボード/VIC-1111: 16K RAMボード

エクスパンション・モジュールのコネクターに接続。

● VIC-1112: IEEE-488 インターフェイス・ボード

IEEE-488を持つ各機器とVIC-1001の接続用。エクスパンション・モジュールのコネクターに接続。

● VIC-1210: 3K RAMパック

VIC-1001のエクスパンション・バス、または、エクスパンション・モジュールの コネクターに接続。

● VIC-1211: スーパー・エクスパンション

176ドット×160ドットの高解像グラフィックおよび音楽演奏を可能にする拡張パック。ファンクション・キーには、使用頻度の高い12のコマンドをアサイン (ユーザーが変更することも可能)。VIC-1211Mは、3 K RAM付き。VIC-100のエクスパンション・バスまたはVIC-1010エクスパンション・モジュールに接続。

● VIC-1212: プログラマーズ・エイド・パック

BASICのプログラミング、プログラムのデバックを大幅に効率化する "TOOL" KIT" のコモドール版。ファンクション・キーには、使用頻度の高い12のコマンドをアサイン (ユーザーが変更することも可能)。VIC-1001のエクスパンション・バスまたはVIC-1010エクスパンション・モジュールに接続。

● VIC-1510 : 専用カラーモニター

コンポジット・ビデオ・インプット、スピーカー内蔵。ボーダー8色、バックグラウンド16色、キャラクター8色。11インチ。モニターケーブルで、VIC-1991のカラービデオ・インターフェイスに接続。

● VIC-1520:ドット・マトリックス・インパクト・プリンター

80桁。印字スピード15ØCPS。グラフィック・キャラクター印字も可能 。VIC - 1ØØ1のユーザーポートに付属ケーブルで、ダイレクトに接続。

● VIC-1521: トラクター・ユニット

ドット・マトリックス・インパクト・プリンター用のトラクター・フィールド・メカニズム。

● VIC-153Ø: カセット・ドライブ

記憶容量 約16gKパイト (C-3g 使用時)。VIC-1gg1のカセット・インターフェイスに接続。

● VIC-1549: シングル・フロッピー・デスク

記憶容量 179Kバイト。VIC-1112 IEEE-488インターフェイス・ボードに接続。

● VIC-131g: ライトペン/VIC-1311: ジョイ・スティック/VIC-1312パドル VIC-1001のコントロールポート、またはマルティブル・コントロール・ボードに 接続。

#### 第3章 VIC-1001の使用方法

VIC-1例1の梱包を解いたら、まず下記の製品が入っていることを確認してください。

VIC-1991パーソナル・コンピューター

電源トランス

家庭用テレビ・アダプター

ユーザーマニュアル

#### 保証書

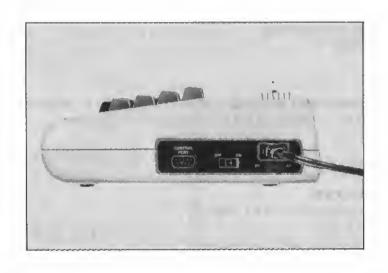
一点でも不足している場合、お買い上げになりました幣社販売店までご連絡下さい。なお、梱包材料は、修理etcの理由により必要になる場合がありますので、保存しておくことをおすすめします。

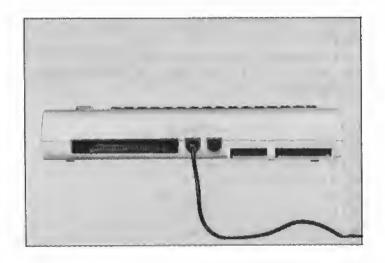
#### 第1節 電源投入の前に

まず、電源トランスのACプラグをコンセントに差し込み、それからジャックをVIC-1901の右側面にあるAC9Vという表示のあるソケットに差し込んで下さい。そのさい、電源スイッチがOFF状態になっていることを確認して下さい。

次に、CRTディスプレイをVIC-1601に接続して下さい。家庭用TVをご使用のさいは、同梱されていますアダプターを家庭用TVのVHFアンテナ端子に接続後、一方のジャックをカラー・ビデオ・インターフェイスに接続して下さい。一方、専用カラー・モニター (VIC-1510)・を御使用の場合は、別売りのモニターケーブル (VIC-1013) を用いて接続して下さい。このさい、TVアダプターは必要ありません。

注意●カセット・ドライブ (VIC-1539) その他の周辺機器、各種オプションと VIC-1691本体との接続方法については、その周辺機器、オプションのマニュアルをお読み下さい。





#### 第2節 電源の投入

まずTVの電源を投入してから、VIC-1991の電源をONにして下さい。 電源スイッチは、右側面に位置しています。この時、ディスプレイの画面が次の表示をすれば、VIC-1991は正常ということになります。

\*\*\*\* CBM BASIC V2 \*\*\*\*
3583 BYTES FREE
READY

注意:上の表示の最下列にある■は点滅カーソルを意味します。本書内の表示の 説明中に何度か出てきますが、常に点滅カーソルの意味です。

#### 第3節 ディスプレイ

#### 1. 画面の文字数

22文字×23行になっています(506文字)。ハイ・レゾリューション・グラフィック(別売りのスーパー・エクスパンダー——VIC-1211が必要となります)の場合は、176ドット×1606ドットになります。

#### 2. カラー

電源投入時、キャラクターはブルー、スクリーンはホワイト、ボーダーはシアンになっています。キャラクターは8色、スクリーンは16色、ボーダーは8色を指定できます。

キャラクターの色指定は、以下の方法となります。CTRL キー (コントロール・キー) を押しながら、 $1 \sim 8$  までのキーを押します。その時点で点滅しているカーソルの色が変ります。

|      | 1 ブラック |
|------|--------|
|      | 2ホワイト  |
|      | 3レッド   |
|      | 4シアン   |
|      | 5マジェンタ |
|      | 6グリーン  |
|      | 7ブルー   |
| CTRL | 8イエロー  |

したがって、COMMODORE VIC-1001の、COMMODOREをレッドで、VICーをシアンで、そして1001をブルーでスクリーン上に表示する場合、その操作の住方は、CTRL キーを押しながら3を押し、いったん離してから、COMMODOREと人力すると、これら9つのキャラクターはレッドで表示されます。 次にCTRL キーを押しながら4を押し、いったん離してから、VICーと入力すると、シアンでVICーと表示されます。最後にCTRL キーを押しながら7を押し、いったん離してから1001と入力すると、1001はブルーで表示されます。なお、プログラム中で、色指定のコマンドが川いられた場合、たとえば

10 PRINT"

の" 内で、CTRL キーと数字キーにより、カラー指定すると、押された数字により以下のシンボルがリバース・フィールド(反転)で表示されます。

|        | グラフィックモード  | カタカナモード |
|--------|------------|---------|
| CTRL 1 | . 🗖        | 11      |
| CTRL 2 | · E        | E       |
| CTRL 3 | . ¥        | ¥       |
| CTRL 4 | · 🔳        | ヲ       |
| CTRL 5 |            | ワ       |
| CTRL 6 | . 🕇        | 1       |
| CTRL 7 |            | 4-      |
| CTRL 8 | - <u>π</u> | $\pi$   |

スクリーンおよびボーダーのカラー・コントロールについては、第2部ソフトウェア編付録Fを参照して下さい。

#### 3-4・キャラクター・コード

VIC-1661のキャラクター・コードに関しては、次ページ以下の表をごらん下さい。

キャラクター・コード表

| ASCH(7 | スキー) | スクリーン | キャラク            |               | ASCH( | アスキー) | スクリーン | キャラ             |               |
|--------|------|-------|-----------------|---------------|-------|-------|-------|-----------------|---------------|
| 10進    | 16進  | ⊐ - F | グラフィック<br>モ ー ド | カタカナ<br>モ ー ド | 10進   | 16進   | □ - F | グラフィック<br>モ ー ド | カタカナ<br>モ ー ド |
| Ø      | Ø    |       | NULL            | NULL          | 38    | 26    | 38    | &               |               |
| 1      | 1    |       |                 |               | 39    | 27    | 39    | •               |               |
| 2      | 2    |       |                 |               | 40    | 28    | 40    | . (             |               |
| 3      | 3    |       |                 |               | 41    | 29    | 41    | }               |               |
| 4      | 4    |       |                 |               | 42    | 2A    | 42    | *               |               |
| 5      | 5    |       |                 |               | 43    | 2B    | 43    | +               |               |
| 6      | 6    |       |                 |               | 44    | 2C    | 44    |                 |               |
| 7      | 7    |       |                 |               | 45    | 2D    | 45    | -               |               |
| 8      | 8    |       |                 |               | 46    | 2 E   | 46    |                 |               |
| 9      | 9    |       | -               |               | 47    | 2F    | 47    | 1               |               |
| 10     | A    | 14    | LF              | LF            | 48    | 30    | 48    | Ø               |               |
| 11     | В    |       |                 |               | 49    | 31    | 49    | 1               |               |
| 12     | С    |       |                 |               | 50    | 32    | 50    | 2               |               |
| 13     | D    |       | CR              | CR            | 51    | 33    | 51    | 3               |               |
| 14     | E    |       | 1               |               | 52    | 34    | 52    | 4               |               |
| 15     | F    |       |                 |               | 53    | 35    | 53    | 5               |               |
| 16     | 10   | ·     | +               |               | 54    | 36    | 54    | 6               |               |
| 17     | 11   |       | カーン             | ル下            | 55    | 37    | 55    | 7               |               |
| 18     | 12   |       | 反               |               | 56    | 38    | 56    | 8               |               |
| 19     | 13   |       | HOME            | HOME          | 57    | 39    | 57    | 9               |               |
| 20     | 14   |       | DEL             | DEL           | 58    | 3A    | 58    | :               |               |
| 21     | 15   |       |                 |               | 59    | 3B    | 59    | :               |               |
| 22     | 16   | _     |                 |               | 60    | 3C    | 60    |                 |               |
| 23     | 17   |       |                 | † · · ·       | 61    | 3D    | 61    | -               |               |
| 24     | 18   | -     |                 |               | 62    | 3 E   | 62    | ĵ               |               |
| 25     | 19   |       |                 |               | 63    | 3F    | 63    | 7               |               |
| 26     | 1A   |       | +               |               | 64    | 40    | Ø     | e               |               |
| 27     | 1B   |       |                 |               | 65    | 41    | 1     | A               |               |
| 28     | 10   |       | L               | y K           | 66    | 42    | 2     | В               |               |
| 29     | 10   |       | _               | ソル右           | 67    | 43    | 3     | С               | 1             |
| 30     | 1.6  |       |                 | ->            | 68    | 44    | 4     | D               |               |
| 31     | 1F   |       | -               | ルー            | 69    | 45    | 5     | E               |               |
| 32     | 20   | 32    |                 | SPACE         | 70    | 46    | 6     | F               |               |
| 33     | 21   | 33    | !               |               | 71    | 47    | 7     | G               |               |
| 33     | 22   | 34    | 4               |               | 72    | 48    | 8     | Н               |               |
| 35     | 23   | 35    | #               |               | 73    | 49    | 9     | 1               |               |
| 36     | 24   | 36    | \$              |               | 74    | 4A    | 10    | ,               |               |
| 37     | 25   | 37    | %               |               | 75    | 48    | 11    | К               |               |

| ASCH | アスキー) | スクリーン | キャラ             | <b>クター</b>    | ASCII ( | アスキー) | スクリーン | キャラ             | クター           |
|------|-------|-------|-----------------|---------------|---------|-------|-------|-----------------|---------------|
| 102  | 1634  | J - F | グラフィック<br>モ ー ド | カタカナ<br>モ ー ド | 10%     | 16進   | ۶ – ۴ | グラフィック<br>モ ー ド | カタカナ<br>モ ー ド |
| 76   | 40    | 12    | L               |               | 116     | 74    | 84    |                 | t             |
| 77   | 4D    | 13    | М               |               | 117     | 75    | 85    |                 | 그             |
| 78   | 4E    | 14    | N               |               | 118     | 76    | 86    | $\boxtimes$     | 3             |
| 79   | 4F    | 15    | 0               |               | 119     | - 77  | 87    | 0               | ラ             |
| 80   | 50    | 16    | Р               |               | 120     | 78    | 88    |                 | リ             |
| 81   | 51    | 17    | a               |               | 121     | 79    | 89    |                 | ル             |
| 82   | 52    | 18    | R               |               | 122     | 7A    | 90    | •               | レ             |
| 83   | 53    | 19    | s               |               | 123     | 7B    | 91    |                 | H             |
| 84   | 54    | 20    | Т               |               | 124     | 7C    | 92    |                 | ワ             |
| 85   | 55    | 21    | U               |               | 125     | 7D    | 93    |                 |               |
| 86   | 56    | 22    | V               |               | 126     | 7E    | 94    | π               | π             |
| 87   | 57    | 23    | W               |               | 127     | 7F    | 95    |                 | 7             |
| 88   | 58    | 24    | X               |               | 128     | 80    |       |                 |               |
| 89   | 59    | 25    | Y               |               | 129     | 81    |       |                 |               |
| 90   | 5A    | 26    | Z               |               | 130     | 82    |       |                 |               |
| 91   | 5B    | 27    | [               |               | 131     | 83    |       |                 |               |
| 92   | 50    | 28    | ¥               |               | 132     | 84    | 127   |                 | ソ             |
| 93   | 5D    | 29    | )               |               | 133     | 85    |       | f               | 1             |
| 94   | 5E    | 30    | t               |               | 134     | 86    |       | f               | 3             |
| 95   | 5F    | 31    | -               |               | 135     | 87    |       | f               | 5             |
| 96   | 60    | 64    |                 |               | 136     | 88    |       | f               | 7             |
| 97   | 61    | 65    | •               | Ŧ             | 137     | 89    |       | f               | 2             |
| 98   | 62    | 66    | Ī               | ッ             | 138     | 8A    |       | f               | 4             |
| 99   | 63    | 67    | F               | ÷             | 139     | 8B    | 1     | f               | 6             |
| 100  | 64    | 68    | H               | ١             | 140     | 8C    |       | f               | 8             |
| 101  | 65    | 69    |                 | +             | 141     | 8D    |       | シフト             | リターン          |
| 102  | 66    | 70    |                 | =             | 142     | 8E    |       |                 |               |
| 103  | 67    | 71    |                 | ヌ             | 143     | 8F    |       |                 |               |
| 104  | 68    | 72    | m               | *             | 144     | 90    |       | ブラ              | ック            |
| 105  | 69    | 73    | N               | 7             | 145     | 91    |       | カー              | ジル上           |
| 106  | 6A    | 74    | N               | - //          | 146     | 92    |       | 反動              | オフ            |
| 107  | 6B    | 75    |                 | Ŀ             | 147     | 93    | 1     | CLR             | CLA           |
| 108  | 6C    | 76    |                 | 7             | 148     | 94    |       | INST            | INST          |
| 109  | 6D    | 77    |                 | _             | 149     | 95    |       |                 |               |
| 110  | 6E    | 78    |                 | 亦             | 150     | 96    |       | 1               |               |
| 111  | 6F    | 79    |                 | 7             | 151     | 97    |       |                 |               |
| 112  | 70    | 80    |                 | 3.            | 152     | 98    |       |                 |               |
| 113  | 71    | 81    |                 | 4             | 153     | 99    |       |                 |               |
| 114  | 72    | 82    |                 | *             | 154     | 9A    |       |                 |               |
| 115  | 73    | 83    |                 | ŧ             | 155     | 98    |       |                 |               |

| ASCII (724-) |      | スクリーン | キャラクター         |          | ASCII (アスキー) |            | スクリーン    | キャラクター          |               |
|--------------|------|-------|----------------|----------|--------------|------------|----------|-----------------|---------------|
| 10%          | 161  | コード   | グラフィック         | カタカナ     | 1000         | 16:        | J - K    | グラフィック<br>モ ー ド | カラカナ<br>モ ー ド |
| 450          | - 00 |       | ₹ - F          | ₹ - K    | 100          | - 04       | CO       |                 | - F           |
| 156          | 90   |       |                | ロンタ      | 196          | 04         | 68<br>69 |                 | +             |
| 157          | 9D   |       |                | ノル左      | 197          | C5         |          |                 | =             |
| 158          | 9E   |       | -              | アン       | 198          | C6         | 70       |                 | -             |
| 159          | 9F   |       | -              |          | 199          | 07         | 71       |                 | X             |
| 160          | AØ   |       | SPACE          |          | 200          | C8         | 72       |                 | *             |
| 161          | A1   | 111   | -              | 0        | 201          | 09         | 73       | 7               | <i>/</i>      |
| 162          | A2   | 98    |                | 1        | 202          | CA         | 74       |                 |               |
| 163          | A3   | 99    |                | ゥ        | 203          | CB         | 75       |                 | E .           |
| 164          | Д4   | 190   |                | I        | 204          | CC         | 76       |                 | 7             |
| 165          | A5   | 101   |                | *        | 205          | CD         | 77       |                 | ^             |
| 166          | A6   | 95    |                | 7        | 206          | CE         | 78       |                 | ホ             |
| 167          | A7   | 103   |                | +        | 207          | CF         | 79       |                 | ₹             |
| 168          | A8   | 1Ø4   |                | 2        | 208          | 00         | 80       |                 | 11            |
| 169          | A9   | 105   |                | ケ        | 209          | D1         | 81       |                 | 4             |
| 170          | AA   | 106   |                | . "      | 210          | D2         | 82       |                 | *             |
| 171          | AB   | 107   | Œ              | <b>F</b> | 211          | D3         | 83       | •               | Ŧ             |
| 172          | AC   | 108   |                | ス        | 212          | D4         | 84       |                 | +             |
| 173          | AD   | 109   | -              | 9        | 213          | <b>D</b> 5 | 85       |                 | ユ             |
| 174          | AE   | 110   | <u> </u>       | 5        | 214          | D6         | 86       | <u> </u>        | 3             |
| 175          | AF   | 111   |                | 0        | 215          | D7         | 87       | Q               | ラ             |
| 176          | 80   | 64    |                |          | 216          | D8         | 88       | +               | 9             |
| 177          | B1   | 97    | -              | ア        | 217          | D9         | 89       |                 | ル             |
| 178          | B2   | 114   |                |          | 218          | DA         | 90       | •               | V             |
| 179          | B3   | 115   | $\blacksquare$ |          | 219          | 09         | 120      |                 |               |
| 180          | 84   | 116   |                | 年        | 220          | DC         | 92       | 38              | 7             |
| 181          | 85   | 117   |                | 月        | 221          | DD         | 121      |                 | ン             |
| 182          | B6   | 102   |                | カ        | 222          | DE         | 106      |                 | *             |
| 183          | B7   | 193   |                | +        | 223          | DF         | 95       |                 | 7             |
| 184          | 88   | 194   |                | 2        | 224          | EØ         | 94       | π               | π             |
| 185          | B9   | 105   |                | 7        | 225          | E1         | 91       |                 | 4             |
| 186          | BA   | 122   |                |          | 226          | E2         | 93       |                 |               |
| 187          | BB   | 123   |                | Ħ        | 227          | E3         | 99       |                 | ゥ             |
| 188          | BC   | 124   |                | ٤        | 228          | E.4        | 100      |                 | I             |
| 189          | BD   | 108   |                | Z        | 229          | E5         | 191      |                 | オ             |
| 190          | BE   | 126   |                | te       | 230          | E6         | 102      |                 | カ             |
| 191          | BF   | 127   | -              | y        | 231          | E.7        | 103      |                 | 辛             |
| 192          | CØ   | 119   |                | 2        | 232          | €8         | 194      |                 | 2             |
| 193          | 01   | 65    | •              | 7        | 233          | E.9        | 105      |                 | 7             |
| 194          | C2   | 66    |                | 7        | 234          | EA         | 106      |                 | *             |
| 195          | C3   | 67    |                | テ        | 235          | EB         | 197      | I E             | (H)           |

| ASCII ( | アスキー) | スクリーン  | キャラ             | クター           | ASCII ( | アスキー) | スクリーン | キャラ             | ララー           |
|---------|-------|--------|-----------------|---------------|---------|-------|-------|-----------------|---------------|
| 10進     | 16准   | € - K  | グラフィック<br>モ ー ド | カタカナ<br>モ ー ド | 10進     | 16進   | ŧ - ۴ | グラフィック<br>モ ー ド | カタカナ<br>モ – ド |
| 236     | EC    | 1 1/18 |                 | ス             |         |       |       |                 |               |
| 237     | ED    | 1 Ø 9  |                 | L             |         |       |       |                 |               |
| 238     | EE    | 110    |                 | 5             |         |       |       |                 |               |
| 239     | EF    | 111    |                 | 0             |         |       |       |                 |               |
| 240     | FØ    | 112    |                 |               |         |       |       |                 |               |
| 241     | F1    | 113    |                 |               |         |       |       |                 |               |
| 242     | F2    | 114    |                 | 1             |         |       |       |                 |               |
| 243     | F3    | 115    | H               | -             |         |       |       |                 |               |
| 244     | F4    | 116    |                 | 年             |         |       |       |                 |               |
| 245     | F5    | 117    |                 | 月             |         |       |       |                 |               |
| 246     | F6    | 118    |                 | B             |         |       |       |                 |               |
| 247     | F7    | 119    |                 | 9             |         |       |       |                 |               |
| 248     | F8    | 120    |                 |               |         |       |       |                 |               |
| 249     | F9    | 121    |                 | ン             |         |       |       |                 |               |
| 250     | FA    | 122    |                 | ⇒             |         |       |       |                 |               |
| 251     | FB    | 123    |                 | t             |         |       |       |                 |               |
| 252     | FC    | 124    |                 | シ             |         |       |       |                 |               |
| 253     | FD    | 125    | P)              | 旦             |         |       |       |                 |               |
| 254     | FE    | 126    | -               | tz            |         |       |       |                 |               |
| 255     | FF    | 94     | π               | П             |         |       |       |                 |               |

#### 第4節:キーボード

#### 1、キーボード配列

VIC-1001には66種類のキーがあり、そのキー配置は次ページの写真の通りです。(JIS準拠)。

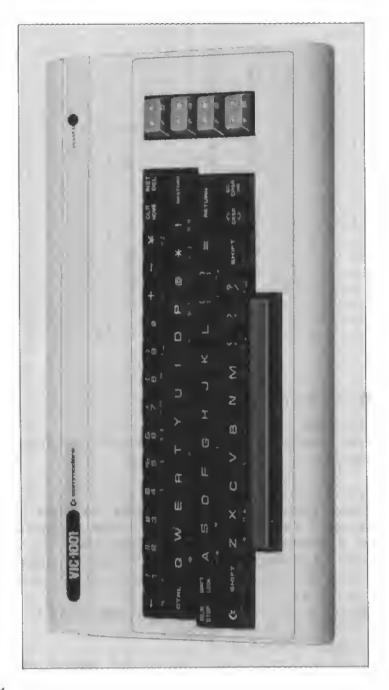
#### 2. キャラクター・モードの選択

VIC-1001は、66種類のキーで英大文字26種、数字10種、特殊文字29種、片仮名 ₹字48種(濁点。、半濁点。を含む)、グラフィック文字50種、漢字3種を入力で とるほか、各種特殊機能をもつようにしています。66種類のキーで、これらのた (さんの文字、機能を入力できるようにするため、VIC-1001では、SHIFT (シフ 上)キーおよび【【\*\*(コモドール)キーを用いて、キャラクター・モードを選択す もようになっています。

VIC-1961のキャラクター・モードは、基本的に次の2種類からなります。

- a. グラフィック・キャラクター・モード
- b、片仮名キャラクター・モード

グラフィック・キャラクター・モードは、英大文字、数字、特殊文字、グラフィ



ック文字からなり、片仮名キャラクター・モードは、英大文字、片仮名文字、漢 字および特殊文字、グラフィック文字の一部から成っています。

#### (1)グラフィック・キャラクター・モード

電源投入時には、キャラクター・モードはグラフィック・モードに設定され、(SHIFT キーニシフト・ロック・キーが押されていなければ)、何もしないで、それぞれのキーを押すと、ディスプレイには図1に担当するキャラクターが表示されます。電源をいったん切って、再度投入し、好きなようにタイプしてみて下さい。

次に、SHIFT キーを押しながら、それぞれのキーを押してみて下さい。ディスプレイには図2に相当するキャラクターが表示されます。

SHIFT キーの代りに、 キーを押しながら、それぞれのキーを押すと、ディスプレイには図3に相当するキャラクターが表示されます。

#### (2)片仮名キャラクター・モード

片仮名キャラクター・モードに移るには、まず、SHIFT キーと優キーを同時に押します。(1)でタイプしたグラフィック文字が片仮名に変わるのがわかるでしょう。

片仮名モードはさらに、片仮名モード I と片仮名モード I に分かれます。片仮名モード I では、図1および図4に相当するキャラクターが表示され、片仮名モード I では、図5および図6に相当するキャラクターが表示されます。

電源投入後、グラフィック・モードから片仮名モードへ移ったさいには、片仮名モードIに設定されています。[SHIFT] キーを押さないで、それぞれのキーを押すと、図1のキャラクターが表示されます。[SHIFT] キーを押しながら各キーを押すと、図4のキャラクターが表示されます(これで最上列の特殊文字をタイプするのに用います)。

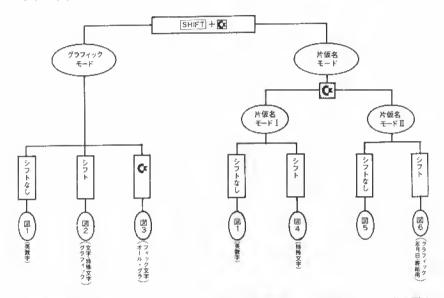
次に、「日キーを押して離すと、片仮名モード」から片仮名モード目に移ります (日キーは、一度押すと、その状態でソフト的にロックがかかります)。片仮名 モード目で、「SHIFT」キーを押さないで各キーを押すと、図5のキャラクター (フル片仮名) が表示されます。「SHIFT」キーを押しながら、各キーを押すと、図6に示すキャラクター (年月日および表組用グラフィック文字) が表示されます。

ここで再び**○** キーを押すと、 片仮名モード [ から片仮名モード ] に移ります。 つまり、 ○ キーが片仮名モード [ とモード ] の切換えスイッチになっています。

片仮名モードからグラフィック・モードに移るには、SHIFT キーと ロキーを同

時に押します。

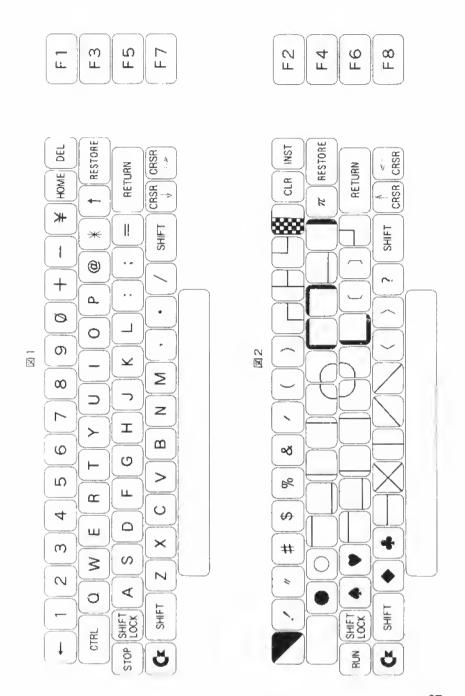
グラフィック・モードと片仮名モード( ${\tt I}$  と ${\tt II}$ )との関係は、次の図のようになります。

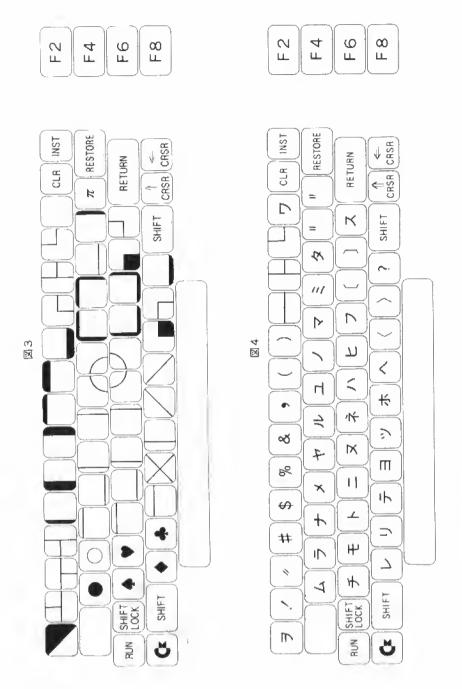


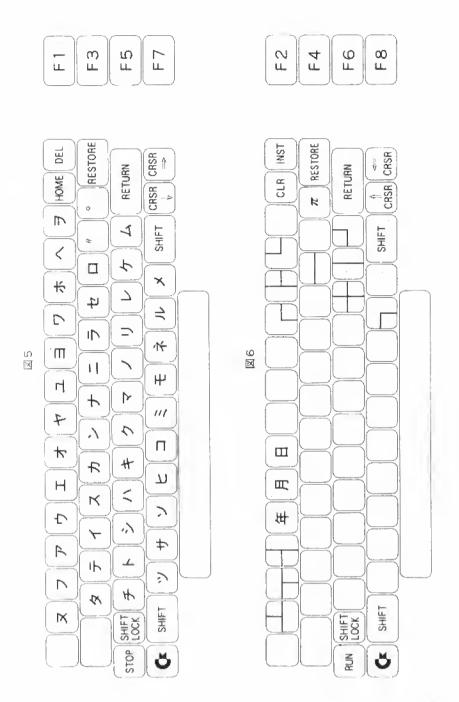
**片仮名の打ち方** プログラム中、片仮名を使うことができるのは、 "で囲んだ中だけです。

例) PRINT "カタカナ" INPUT "キカン"; K REM "オセロ" A \$="コモドール"

したがって、一般に、プログラム中で片仮名を使いたいときは、次のようにします。まず、プログラム中で片仮名を使うかどうかを決め、グラフィック・モードから片仮名モードに変換します。そして、たとえば10 PRINT "カタカナ"と打とうとして、「1を押して、ヌが表示されたら、「2 キーを押します。そしてそのまま、シフトなしでタイプすれば 10 PRINTとタイプできます。それから、「SHIFT」キーを押しながら「2 を押します。そのあと、「2 キーを押して離してから、「カタカナとタイプします。片仮名が打ち終ったら、「2 キーを押して離し、「SHIFT」キーを押しながら、「2 を押します。これで、10 PRINT "カタカナ"とタイプできたことになります。







#### 3 ・ 特殊キーの働き

- (1) **CTRL キー (コントロール・キー)** このキーは、次の3つの用途に用います。
  - a. キャラクターのカラー・セット
  - b. リバース・フィールド(反転文字)のセットおよび解除
  - c、実行スピードのダウン

キャラクターのカラー・セットは、第4章第3節3のカラーの項を参照して下さい。 リパース・フィールドのセットは、CTRL キーを押しながらRを押します。リバース・フィールドの解除は CTRL キーを押しながら Øを押します。

画面上にABCをリバース・フィールドで表示する場合

?"CTRL | R CTRL | ABC CTRL | | Ø CTRL | 1"

と押します。この時点で画面は、?" R ABC — "と表示され、 RETURN キーが押されると、 ABCがリバース・フィールドで表示されます。

なお、カタカナ・モードの場合は、表示は、?" [RIABC|メーとなります。

また、CTRL キーの3番目の使い方は以下の通りです。プロ々ラム実行中、このキーを押し続けている間、プログラムの実行スピードをおとすことができます。同様に、プログラムをLIST中にこのキーを押すことにより、スクロール・スピードをおとすことができます。

(2) RUN キー (ラン/ストップ・キー)

実行中のプログラムを停止する時、そして、プログラムをカセット・ドライブ からLOADするのを中止させる時に用います。SHIFT (または ♥ ) キーを押しながら、このキーを押すことにより、カセット・ドライブから最初にみつかった プログラムをLOAD開始します。

- SHIFT LOCK SHIFT 状態のモードに保ちます。
- (4) (4) (4) キー(コモドール・キー)他のキーと併用します。(その働きについては、前項を参照して下さい)
- (5) SHIFT キー (シフト・キー) 他のキーと併用します。(その働きについては、前項を参照して下さい)

#### (6) CLR +- (クリヤ/ホーム·キー)

このキーを押すと、カーソルがホーム・ポジション(左上端)に戻ります。 SHIFT キーと同時に押すと、画面をクリヤーし、そしてカーソルをホーム・ポジションに戻します。

#### (7) INST キー (インサート / デリート・キー)

このキーを押すと、カーソルの左側にあるキャラクターを1字削除し、それより右の文字が左へ詰められます。リピート機能がついています。[SHIFT] キーと同時に押すと、カーソルの位置とその左側のキャラクターの間にスペースを入れ、キャラクターの挿入を可能にします。

#### (8) RESTORE キー (レストア・キー)

プログラム実行中、何らかの理由によりプログラムが暴走した場合、RUN STOP キーとともにこのキーを押すことにより、メモリー中のデータを失うことなく実行を停止させることができます。

#### (9) RETURN キー (リターン・キー)

カーソルを次列の左端に移します。カーソルが最下段にある場合は、画面をス クロールします。

#### 10 CRSR キー (カーソル・アップ/ダウン・キー)

このキーを押すと、カーソルを1列下に下げます。SHIFT キーと併用した場合は、1列上に上げます。リピート機能がついています。

#### (11) CRSR キー (カーソル・レフト/ライト・キー)

このキーを押すと、カーソルを1キャラクター分石へ移動させ、SHIFT キーと併用した場合は、左へ移動させます。

#### (12) f1~f8キー(ファンクション・キー)

スーパー・エクスパンダー(VIC-1211)またはプログラマース・エイド・パック(VIC-1212)を使用することにより、12種類の機能を持たせることが可能になります。VIC-1211または1212をご使用にならない場合でも、プログラムにより、一定のファンクションを持たせることが可能です。なお、VIC-1211、1212をご使用の場合はそれぞれのマニュアルをご参照下さい。

#### 第4章 BASICによるプログラミング

この章は、VIC-1001の操作方法に習熟していただくために、特に初心者の方を対象に、BASIC (ベーシック)言語でプログラムをどのように書き、編集するか、またそれをどのように実行させるかなど、基本的な事項の説明をおこないます。 VIC-1001のBASIC言語であるとCBM BASIC V2 (バージョン2)の詳しい説明については、第2部ソフトウェア編を参照して下さい。

#### 第1節 BASICとは

BASIC (Beginner's All-purpose Symbolic Instruction Gode) は、その名の通り、初心者向けコンピューター言語として米国で開発され、今ではほとんどのパーソナル・コンピューターに、BASIC言語がとりいれられるようになりました。対話形式を基本としたその使いやすさが受け入れられており、CBM BASIC V2は、使われる方々の立場に立って特に開発されたものです。

BASIC言語は、パーソナル・コンピューター・メーカーや、その機種によって 命令の書き方や機能が少しずつ異なっていますので、必ず第2部のソフトウェア 編も読んで下さい。

#### 第2節 プログラムの書き方

VIC-1001に電源を入れると、

\*\*\*\* CBM BASIC V2 \*\*\*\*
3583 BYTES FREE
READY.

と表示されます(■は、点滅カーソルを意味します)。ディスプレイ上に点滅カーソルが出ている場合、コンピューターはREADY状態にあり、人力待ちを意味します。たとえば、キーボードからでたらめな文字(数字はこのさい含めないで下さい)を人力し、「RETURN」キーを押して下さい。

?SYNTAX ERROR READY. とディスプレイに表示されるはずです。SYNTAX ERRORとは、コンピューターの理解できない文章が人力された場合に表示されるメッセージです。この他にもエラー・メッセージは何種類もあります。どのような場合にどのようなエラー・メッセージが表示されるかは、第2部付録Dを参照して下さい。

VIC-1001に何らかの動作をさせるのには、(1) ダイレクト・モード (2) プログラム・モードの 2 つの方法があります。ここでは、この 2 つのモードを BASICの面から説明します。

#### (1)ダイレクト・モード

ライン・ナンバー(行番号――プログラム・モードの項を参照して下さい)をつけずに、BASICコマンドを直接入力することにより、コンピューターに命令を与えることをダイレクト・モードと呼びます。たとえば、

#### PRINT "GOOD MORNING" RETURN

と入力して下さい。その時、ディスプレイは次のように表示します。

PRINT "GOOD MORNING" GOOD MORNING

READY.

4.

●注意● 入力の終りをコンピューターに知らせるために、入力の最後には必ず RETURN キーを押します。 RETURN キーが押されないと、コンピュー ターはいつまでも入力があるものと考え、実行に移りません。

ダイレクト・モードは、コンピューターに命令を記憶させずに、直接結果を得よ うとする場合に用います。たとえば、

PRINT 456 \* 346/3

52592

READY.

P

PRINT SIN (3 \* \pi / 360 \* 69)

READY.

1

PRINT TI\$ 999152 READY.

-

VIC-1661ではBASICのいろいろなコマンドを省略形を用いて書くことができます。そのうち、最も頻繁に用いられるものに?があります。これは、PRINTの代用になります。省略形を用いて、上記の入力をする場合、以下のようになります(省略形については第2部ソフトウェア編付録Bを参照して下さい)。

7 456 \* 346/3 52592

READY.

? SIN (3 \* π/360 \* 60)

1

READY.

? TI\$

000325

READY.

また、変数を1つ1つ定義し、最後に変数を用いた数式を書いて値を求めることもできます。たとえば、球の体積を求める場合、半径が10であるならば、

R=10

READY.

2 4/3 × π × R † 3

4188.79@2

READY.

また、直方体(20×30×40) の体積を求める場合、

A=20:B=30:C=40 READY. ? A \* B \* C 24000

READY.

ダイレクト・モードでは、上で説明したように、結果はすぐに得られますが、 コンピューターの特長である「人間の命令した手順通り、複雑な計算を何回もおこ なったり、大量のデータを自動的に処理する」ことはできません。

#### (2)プログラム・モード

ダイレクト・モードではライン・ナンバー(行番号)をつけませんでしたが、プログラム・モードではライン・ナンバーをつけることにより、コンピューターに実行の順番を指示します。 CBM BASICでは、1ライン・ナンバーごとに88字(ライン・ナンバーおよびスペース、そして RETURN キーを含めて)までを人力することが可能です。コンピューターはRUNという入力があると、ライン・ナンバー順に実行開始します。BASICの命令を順序よく考えて、組み合わせ、処理の手順として記憶させていく作業をプログラミングと呼びます。ここでは、球の体積を求めるプログラムを作ってみましょう。半径をRとします。

167 "ハンケイ? " RETURN 26 NPUT R RETURN 367 "タイセキ="; RETURN 46? 3/4\*π\*R † 3 RETURN

と人力した後、LISTと入力して下さい(LISTの後にも RETURN) キーを押すことを忘れないで下さい)。ディスプレイは、以下を示します。

- 10 PRINT "ハンケイ?"
- 20 INPUT R
- 30 PRINT "タイセキ=":
- 40 PRINT3/4 \* π \* R † 3

お気付きのように、省略形を入力時に用いたにもかかわらず、LIST(入力した プログラムの内容を確める場合に用いるコマンド) すると、BASICコマンドが、 完全な形で表示されます。また ライン・ナンバーの後にスペースを入れなくても、 LIST時には、スペースが入り、読みやすくなります。

なお、プログラミングを開始する前には、メモリーの内容をクリヤーするため にNEWというコマンドを用います。ただし、いったんNEWというコマンドを 入れますと、画面上には残っていても、メモリー中からは除去されてしまいます ので、気をつけて下さい。

## 第3節 プログラムの修正

まず、前記のプログラムが実際に、どのように動くかを確めるために、RUN RETURN と入力して下さい。

10 PRINT"ハンケイ?"
20 INPUT R
30 PRINT"タイセキ=";
40 PRINT3/4\*π\*R†3
PEADY.
RUN
ハンケイ?
?■

となります。この時点で、半径の値を入れます。たとえば半径=15として、15 |RETURN| と入力すると、画面は

ハンケイ? ? 15 タイセキ=7952.1564 READY.

となり、体積が求められました。ところがこのプログラムの場合、次にたとえば 半径が20の球の体積を求めようとすると、ふたたびRUN RETURN と入力する 必要があります。この手間を省くためにはプログラムの修正をする必要がありま す。また、画面をいったんクリヤーしてから、実行に移るという修正も加えてみ ることにします。

まず、プログラムをLISTして下さい。そして、点滅しているカーソルの位置から以下を入力して下さい。

5 7 " SHIFT CLR HOME "

そして再度LISTと人力して下さい。画面には、以下が表示されるはずです。

5 PR!NT "♥"

10 PRINT "ハンケイ?"

20 INPUT R

30 PRINT "タイセキ=";

40 PRINT3/4 \*π \*R † 3

50 GOTP10

READY.

お気付きのように、後に加えた2行のラインが、ライン・ナンバー順に並び変えられました。このように、後に加えたプログラムも、ライン・ナンバーにより必要なところへ挿入が可能になります。

そこで、この修正されたプログラムをRUNし、15を人力して下さい。すると、 画面がいったんクリヤーされ、スクリーンの左端から、

ハンケイ?
? 15
タイセキ=7952.1564

?SYNTAX ERROR IN 50 READY.

. .

と表示されます。

まず、ライン・ナンバー 5 を加えたことにより、キーボードの項で説明しましたように、  $\boxed{\text{SHIFT}}$  +  $\boxed{\text{CLR}}$  により、いったん画面がクリヤーされました。

次に、\*?SYNTAX ERROR IN50″というメッセージにより、ライン・ナンバー50に文法上の誤りがあったことがわかります。そこで、LIST50と人力して下下さい。すると、画面は以下のようになります。

LIST 50

50 GOTP 10 READY.

p |

ここで、BASICに登録されていないGOTPという言葉があることが判ります。し

たがって、カーソル・コントロール・キー ( $\widehat{\text{CRSR}}$  、 $\widehat{\text{CRSR}}$  、 $\widehat{\text{CRSR}}$  、 $\widehat{\text{HOME}}$  および  $\widehat{\text{DEL}}$  )を用いて、修正をおこないます。

上の場合は、SHIFT キーと CRSR を用いて、カーソルを50 GOTP 50 の位置まで持って行き、その次に CRSR キーで、カーソルをPを上に移し、その位置でOを打ち、 RETURN キーを押します。そして、カーソルをREADYの下の行まで持って行き、LISTと入力して下さい。ここで、ライン・ナンバー50が以下のように訂正されている事を確認して下さい。

#### 50 GOTO 10

- ●注意 1 訂正後、RETURN キーを使用せずに、カーソル・コントロール・キーを用いて下の行に移った場合、その行は訂正されません。
- ●注意 2 プログラムのライン・ナンバーを10とか100おきにするのは、後になって行を追加する場合があるからです。ライン・ナンバーは正の整数である必要があり、最大63999までになっています。

つぎに、このプログラムをRUNして、今度は正確に動くことを確認して下さい。 半径の値を変える事により、次々に体積を瞬時に計算し、ディスプレイに表示します(このプログラムの実行を停止させるには、[RETURN] キーを押します)。

次に、簡単に、ライン・ナンバーごとに、どのような処理がなされるのか説明します。

まずライン・ナンバー5では、前に述べましたように、画面をクリヤーし、カーソルをホーム・ポジションに持っていくという処理がなされます (PRINT "の間にカーソル・コントロール・キーを用いれば、好みの場所に文字を表示できます。ライン・ナンバー5を修正して、実際に研究してみて下さい。)ライン・ナンバー19で、ハンケイ?という文字列をディスプレイ上に表示させます。ライン・ナンバー20では、Rという変数に対して、数字が入力されるのを待ちます (数字以外が入力されると、コンピューターは?REDO FROM START…最初からやり直して下さい……という表示をします)。ライン・ナンバー30では、10と同様にタイセキ=という文字列をディスプレイ上に表示させます。その行の終りにある;は、次の表示を、タイセキ=の行どなりに表示させます。その行の終りにある;は、次の表示を、タイセキ=の行どなりに表示されます。

ライン・ナンバー40では、人力されたRの値を実際に数式にあてはめ、その結果を、上記タイセキ=の右側に表示します。PRINTの後に、""が無い事に注目して下さい。""をつけた場合、数式そのままを表示します。

例1: ハンケイ?

7 15

タイセキ=7952.1564

READY.

例2: ハンケイ?

? 15

タイセキ=

7952.1564

ある行全体を削除する場合は、該当する番号を入力し、RETURN キーを押します。たとえば、前の例の中のライン・ナンバー5を削除する場合は、

#### 5 RETURN

と人力します。

また、LISTコマンドは以下の使い方があります。

1. ある1行(ライン・ナンバー10とします)をリストする場合

LISTIM RETURN

2. ライン・ナンバー10~150までをLISTする。

LISTIW-150 RETURN

3. ライン・ナンバー150までをLISTする。

LIST-150 RETURN

4. ライン・ナンバー15Ø以降をLISTする。

LISTIS#- RETURN

スクロール(画面のせり上がり)をストップさせるには、 $\begin{bmatrix} \mathrm{RUN} \\ \mathrm{STOP} \end{bmatrix}$  キーを用い、 $\Phi$ っくりスクロールさせたい時は $\begin{bmatrix} \mathrm{CTRL} \\ \mathrm{TRL} \end{bmatrix}$  キーを用います。

あるライン・ナンバーに修正を加え、そのライン・ナンバーの文が23字より長くなる場合、自動的に行間隔があけられます。たとえば、

10 PRINT" 1 2 3 4 5 6 7 8 9 0"
20 PRINT" ABCDEF"

とあり、ライン・ナンバー1 $\emptyset$ の123456789 $\emptyset$ の代りに123456789 $\emptyset$ 12345とする場合、 カーソルを $\emptyset$ の後の''の位置へ戻し、123と追加するわけですが、3を入力した時点で、ライン・ナンバー2 $\emptyset$ は自動的にスクロール・ダウンします。

## 第4節 スクリーン・エディター

プログラムの編集に関するまとめおよび補足をします。

- 1) 1ライン・ナンバーを書き終ったあと、そして修正し終った後、必ず RETURN キーを押して下さい。たとえば1月PRINT "ABC"と入力し、RETURN キーを押さずに、CRSR キーを用いて下段にカーソルを移動し、LISTしても、何もディスプレイされない事で、RETURN キーの必要性は、おわかりいただけると思います。一方、あるライン・ナンバーを書き終えていない場合は、カーソルが画面の行端にあったとしても、その次の行には自動的にカーソルは移動しますから、RETURN キーは押さないで下さい。RETURN キーが押されると、入力が終了したとみなされます。
- 2) あるライン・ナンバーを持つ行全体を消す場合は、そのライン・ナンバーを入 力し RETURN キーを押します。
- 3) あるライン・ナンバーと同じ内容の行を違ったライン・ナンバーとともに用いたい場合、そのライン・ナンバーの上にカーソルを移動し、新しい番号を入力し、RETURN キーを押すことにより、新しい行を発生させることができます。たとえば

#### 10 PRINT "HELLO":

と入力して下さい。そして、1の上にカーソルを戻し、20 RETURN 、そして次にカーソルを2の上に戻し、30 RETURN と入力し、LISTしてみて下さい。

- 10 PRINT"HELLO";
- 20 PRINT"HELLO";
- 30 PRINT"HELLO";

となります。

# 第6章 カセット・ドライブの使い方

VIC-1991では、電源を切るとメモリー内のプログラムおよびデータは消えてしまいます(ROMに書き込まれているものは消えません)。したがって、そのプログラムやデータを再度用いる場合には、あらかじめそれらを外部記憶装置に記憶させておきます。VIC-1991の外部記憶装置には、カセット・テープおよびフロッピー・ディスクがあります。VIC-1999シリーズの初心者用システムおよび標準システムでは、カセット・テープを用いますので、ここではカセット・ドライブ(VIC-1539)の使い方について説明します。まず、カセット・ドライブのマニュアルに従って、カセット・ドライブをVIC-1991に接続し、所定の初期テストおよび録音・再生チェックをおこなったうえで、以下をおこなって下さい。

次の簡単なプログラムをキーボードから入力して下さい。

10 FOR L= 1 TO 20 20? "HELLOW VIC"; 30 NEXT I

このプログラムをカセット・テープに記憶させるには、SAVEというコマンドを用います。

## 第1節 プログラムのSAVE

まず、SAVEする前に、このプログラムに名前をつける必要があります。プログラム名は、最大84文字使用できますが、実際にテープに書き込まれるのは、16文字までです。したがって、上のプログラム名をABCDEFGHIJKLMNOPQRSTUVWXYZという26文字としてSAVEしたとしても、テープ上にはPまでしか書き込まれません。つまり、もし、2番目のプログラムが、ABCDEFGHIJKLMNOPZZZという名称でSAVEされた場合、後にコンピューターは、これら2つのプログラムとも、プログラム名は、ABCDEFGHIJKLMNOPであると判断します。

さて、実際にこのプログラムにTESTという名称をつけてSAVEすることにします。 SAVE "TEST" [RETURN] と入力して下さい。

aticiti (C.

PRESS RECORD & PLAY ON TAPE

と表示されます。画面の指示にしたがい、カセット・ドライブのRECORDボタンを押しながら、PLAYボタンを押すと、画面に、

OK

#### SAVI NG TEST

が表示され、SAVEが終了すると、画面にREADY. が表示され、カーソルが点滅します。

●注意● 実際に、SAVINGという表示がスクリーンに出てから約16秒間は、カセット・ドライブのモーターの回転が安定化を待つために録音は開始しません。約16秒経過後、ヘッダーと呼ばれるものを書き込みます。そして録音終「後、エンド・マークを書き込みます。録音開始まで約16秒間ありますので、問題はないと思いますが、クリアフィーダー(磁性体の塗ってない部分)が異常に長い場合、ヘッダーがカセット・テープに録音されず、後に再生できない場合があります。

## 第2節 プログラムのVERIFY

上記の原因の他にも、うまくSAVEできないことになる原因はあります。たとえば、テープの傷、RECORDボタンの押し忘れが典型的なケースです。これらを防止するために、SAVE終了後、VERIFYしておく習慣をつけて下さい。

前例のプログラム \*TEST" をSAVE終了後、カーソルが点滅し始めた時にいったん巻き戻し、キーボードからVERIFY TESTと人力します。すると、画面には、

#### PRESS PLAY ON TAPE

と表示されますので、表示にしたがい、カセット・ドライブのPLAYボタンを押して下さい。

画面には次のようなメッセージが出てきます。

OK SEARCHING FOR TEST FOUND TEST VERIFYING OK

READY.

なお、前に述べた理由により、 SEARCHING……のメッセージとFOUND TEST のメッセージの間に約10秒間の間隔があります。

上記のメッセージが表示されれば、プログラムは正しくSAVEされました。

もし、SAVEが正しくなされていない場合は、次のような結果となります。

- もし、ヘッダーがクリアフィーダーの終わらないうちに書き込まれた場合、 ヘッダーが検知されないことにより、FOUND TESTのメッセージがいつ まで待っても出てきません。
- 2) もし、何らかの理由により、うまくSAVEされていない場合には、最後の OKの代りにVERIFY ERRORというメッセージが表示されます。

以上でおわかりと思いますが、VERIFYとは、コンピューターのメモリー中にあるプログラムと、指示されたプログラム名で記録されたカセット・テープ内のプログラムを比較し、一致しているかどうかを「確認する」ことをいうのです。

ここで、いったんコンピューターのメモリーをクリヤーするためにNEWというコマンドを入力して下さい。その次にLISTと入力し、メモリーがクリヤーされたことを確認して下さい。

## 第3節 プログラムのLOAD

ここでは、プログラムをカセット・テープからLOADする方法について説明 します。

次のように、キーボードから入力して下さい。

LOAD "TEST"

画面に次のようなメッセージが表示されます。

#### PRESS PLAY ON TAPE

指示にしたがい、カセット・ドライブのPLAYボタンを押すと、画面に次のよ

うなメッセージが出ます。

OK

SEARCHING FOR TEST FOUND TEST LOADING

READY.

これで、プログラムのLOADが終了したことになります。

また、その時次の実験をしてみて下さい。LOAD "XYZ" と人力し、画面の指示にしたがい、カセット・ドライブのPLAYボタンを押して下さい。画面には次のようなメッセージが表示され、モーターは回転し続けるはずです。

#### SEARCHING FORXYZ FOUND TEST

おわかりのように、指定されたプログラム名以外のプログラムは、その名前だけを表示し、LOADはしません。

# 第4節 データのLOADおよびSAVE

この項を説明するには、2本のテープが必要になります。以下の説明をお読み になる前にまず、2本のテープをご用意下さい。

以下のプログラムをキーボードから入力し、前項で説明した要領でPGM-1というプログラム名でSAVEして下さい。

- 10 OPEN 1, 1, 1
- 20 FOR J=1 TO 20
- 30 PRINT#1, "LAM A VIC!"(?を用いないで、PRINT…として下さい)

- 40 NEXT J
- 50 CLOSE 1
- 60 PRINT "REWIND YOUR TAPE AND THEN PRESS A SPACE KEY"
- 70 GET A\$: IF A\$=""THEN70(クオート"を続けて2回タイプして下さい)
- 80 OPEN 1
- 90 FOR J=1 TO 20
- 199 INPUT#1, X\$
- 110 PRINT X\$
- 120 NEXT J
- 130 CLOSE 1
- 140 PRINT "WONDERFUL!!"

そのカセットを巻き戻し、PGM 1と書いたラベルをはっておいて下さい。 別のカセットを入れて下さい。完全に巻き戻されていることを確認後、上のプログラムをRUNさせて下さい。

#### PRESS RECORD & PLAY ON TAPE

という表示が出ますから、指示にしたがいカセット・ドライブのRECボタンを 押しながらPLAYボタンを押して下さい。

すると、OKが表示され、その約25秒後に、以下が表示されます。

#### REWIND YOUR TAPE AND THEN PRESS A SPACE KEY

指示にしたがい、REWボタンを押してテープを巻き戻し、キーボードのスペース・キーを押します。すると、

#### PRESS PLAY ON TAPE

と表示されますので、指示にしたがいカセット・ドライブのPLAYボタンを押して下さい。すると、OKと表示が出て、約25秒待つとIAM AIVIC!という文章が20行プリントされ、最後にWONDERFUL!!と表示し、READY状態となります。

●注意● この時、I AM A VIC!を16行プリントし、テープがいったん止まりますが、これはバッファーがいっぱいになるためで、支障はありません。

ここで、テープを巻き戻し、DATA 1と書いたラベルをはっておいて下さい。

以上で、一連の作業は終りました。

PGM 1とマークされたテーブがプログラム・テープであり、DATA 1とマークされたテーブがデータ・テープです。ここで、ちょっとした実験をしてみましょう。DATA 1とマークされたテープをカセット・ドライブに入れ、 $\overline{SHIFT}$  キーを押しながら $\overline{RUN}$  キーを押して下さい。 画面に

#### PRESS PLAY ON TAPE

が表示されます。この表示にしたがい、カセット・ドライブのPLAYボタンを押して下さい。

#### 画面には、

OK

#### SERCHING FOUND

と表示されるだけで、LOADINGというメッセージは出てきません。つまり、データ・テープは、プログラム・テープとは違ってLOADすることはできません。 以下でライン・ナンバー順に何がなされているかを説明します。

#### ライン・ナンバー18

ロジカル・ファイル1 (最初の1) をデバイス・ナンバー1 (2番目の1…… カセット) 用に、しかも書き込み用(最後の1) にOPENする(開く)。

#### ライン・ナンバー20~40

テープに I AM A VIC!を20回、書き込む。

#### ライン・ナンバー50

ロジカル・ファイル 1 をCLOSEする (閉じる)。

#### ライン・ナンバー60~70

テープを巻き戻すメッセージを表示し、スペース・キーが押されるまで待つ。 メッセージは、、スペース・キーを押す "とありますが、実際には、何でもキーが 押されれば、次のライン・ナンバーに進みます。

#### ライン・ナンバー80

ロジカル・ファイル1をOPENする。

#### ライン・ナンバー90~120

テープから I AM A VICを20回読み取り、表示する。

#### ライン・ナンバー130

ロジカル・ファイル1をCLOSEする。

#### ライン・ナンバー140

WONDERFUL!! を表示する。

もう少し詳しく説明しましょう。ライン・ナンバー10は、カセット・ドライブ (周辺機器には必ず、デバイス・ナンパーが割り当てられており、VIC-1001 の 場合、カセット・ドライブが1となっています) 用に、チャネルを、書き込み用にO PEN したわけです。つまり、最初の1はOPEN したいチャネル番号を示し、次の1はデバイス・ナンバーを示し、最後の1は、そのチャネルを、書き込み用にOPENすることを意味します。

VIC-1001では、チャネル(正確にはロジカル・ファイルと呼びます )を10本 までOPENすることが可能です。10本以上のチャネルがOPENされると、

#### ? TOO MANY FILES ERROR

となります。

また、ロジカル・ファイル・ナンバーは1から255までの間の整数であれば、いくつでもかまいません。256以上の数字を用いると、

#### ? ILLEGAL QUANTITY ERROR

となります。

最後の1は、書き込み用にチャネルをOPENすることを意味しています(テープから、読み取る場合は、%になります。――ライン・ナンバー8%でもう少し説明します)。

まとめをかねて、もう一度説明しますと、データをカセット・テープやフロッピー・ディスクに書き込むには、

#### OPEN A, B, 1

とし、その場合 Aは、ロジカル・フィイル・ナンバーであり、Bがデバイス・ナンバーとなります。カセットの場合、Bは1です。

ライン・ナンバー20から40では、OPENしたロジカル・ファイルに20回のI AM A VIC!を送り込んでいます。そしてライン・ナンバー50で、先にOPEN されたロジカル・ファイル I をCLOSEします。この場合、OPENしたロジカル・ファイル・ナンバーと同じナンバーをCLOSEしないと、ライン・ナンバー80でエラーとなります。つまり、いったんOPENしてあるロジカル・ファイルをCLOSEせずに、再度OPENすると、

#### ? FILE OPEN ERROR IN 80

となります。

ライン・ナンバー70以下では、書き込まれたデータの読み込みをします。ライン・ナンバー80で、先ほどと同様にロジカル・ファイル1をOPENしています。このライン・ナンバーは、実際には、

#### 80 OPEN 1. 1. 0

であるべきものです。しかし、特に指示しないかぎり、デバイス・ナンバーは1であり、ロジカル・ファイルは読み取り( $\emptyset$ —これをセカンダリー・アドレスと呼びます)用にO P E N されます。ライン・ナンバー10で、デバイス・ナンバーおよびセカンダリー・アドレスを定義したのは、CBM BASICでは常にOP E N  $\langle$  ロジカル・ファイル・ナンバー $\rangle$ 、 $\langle$  デバイス・ナンバー $\rangle$ 、 $\langle$  セカンダリー・アドレス $\rangle$  の順に書く必要があり、ライン・ナンバー10ではセカンダリー・アドレス $\emptyset$ でなく、1 であるため、必然的にデバイス・ナンバーも必要になったわけです。

前に、プログラムのLOADの所で、PGM 1をLOADする場合、

LOAD "PGM 1"

としましたが、これも実際には

LOAD "PGM 1" , 1

であったわけで、この1は、デバイス・ナンバーです。したがって、1であるため、省略ができたわけです。

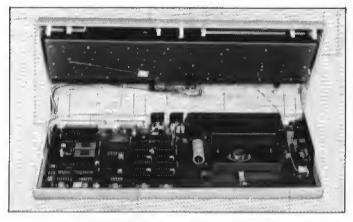
# 第6章 ハードウェアの説明

## 第1節 概論

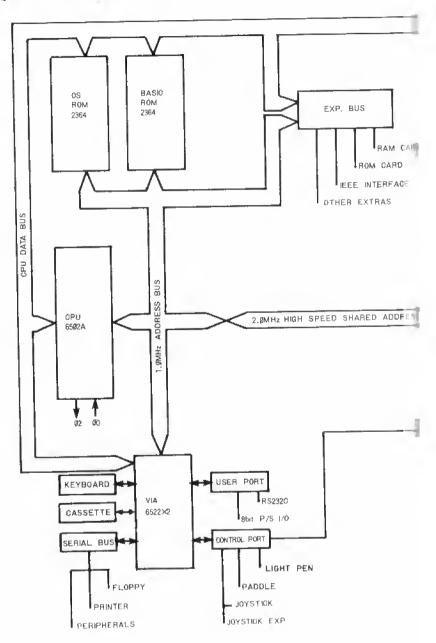
VIC-1001は、CPUにMOS TECHNOLOGY社 (コモドールの半導体部門)製のMPS-6502Aが使用されているパーソナル・コンピューターです。このクラスでは最も充実したインターフェイスを内蔵しており、システムの拡張が多様かつ容易に可能です。以下では、VIC-1001のハードウェアについて説明します。

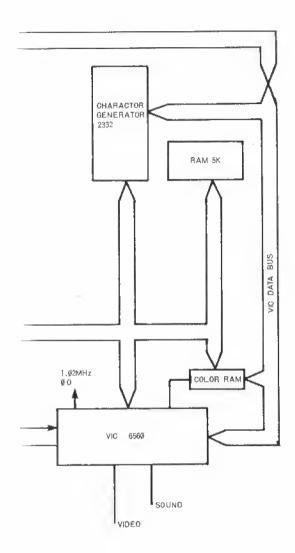
## 第2節 外観





# 第3節 システム構成





## 第4節 機能仕様

(I)CPU

使用CPU : MPS-6502A

CLOCK 周波数: 1,0227MHz (14,31818MHzを14分周)

RESET : パワー・オン

割込み : カード・エッヂに開放

(2)ROM

使用ROM: MPS 2364×2

: MPS2332×1

容量 : 20Kバイト

: 最大32Kバイトまで拡張可能

(3)RAM

使用RAM : MPS2114×10

容量 : 5 K バイト

最大32Kバイトまで拡張可能

(4)CRTインターフェイス

使用コントローラー: MPS 6560

スクリーン構成: 22桁×23行

文字構成 : 8×8ドットマトリックスキャラクター128種類

8×8グラフィックキャラクター64種類

グラフィック機能: 176×160または88×160

カラー機能 : キャラクター8色

バックグラウンド16色

ボーダー8色

カーソル : リバース・プリンク・カーソル

その他 : リパース

(5)カセット・インターフェイス

方式 : コモドール方式

ボーレート : 500ボー

(6)サウンド・ジェネレーター

構成 : サウンド・ジェネレーター×3(別個にコントロール可能)

: ホワイト・ノイズ・ジェネレーター×1

音域 : 128Hz~16KHz

(7)ユーザー・ポート

方式: 8ビット・パラレル/8ビット・シリアル人出力兼用

(8)シリアル・インターフェイス

方式: コモドール方式

 $35 - \nu - \nu$  : 50/70/134.6/150/300/600/1200/2400/3600

(9)キーボード

方式 : ソフトウェア・スキャン キー : JIS 準拠フルキーボード

> 4ファンクション・キー、カーソル・コントロール・キー、 コントロール・キー、コモドール・キー、レストア・キー

(10)電源

電源電圧 : AC1ØØV±1Ø%、5Ø/6ØHz

消費電力 : 18W

11使用条件

使用温度 : 0℃~40℃

使用湿度 : 20%~80% (ただし結露しないこと)

保在温度 : −5℃~60℃

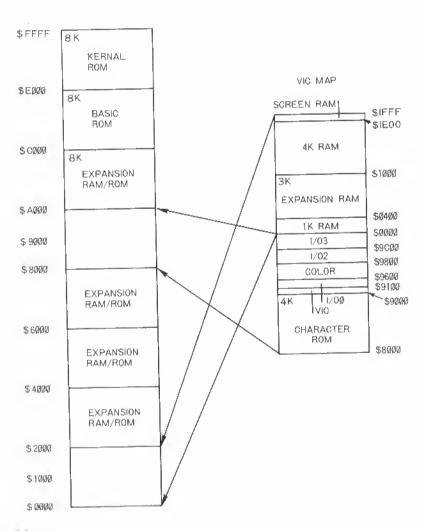
(12)外形寸法 : 4Ø4(W)×2Ø4(D)×74(H)mm

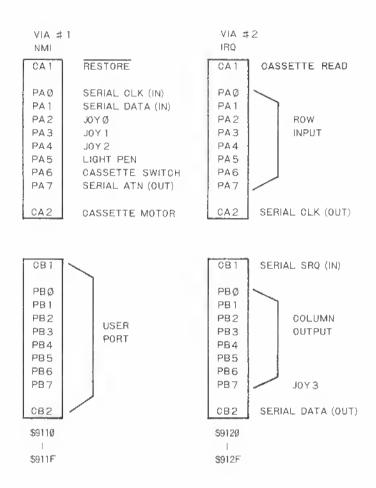
(13)重量 : 1..8kg

# 第5節 メモリー・アドレス・マップ

VIC-1 $\emptyset\emptyset$ 1のメモリーには、BASICのインタープリターが収納されているマスクドROM(MPS2364)と、スタテックRAM(MPS2114)が使われており、ROM2 $\emptyset$ Kバイト、RAM5KKバイトが実装されています。ROMおよびRAM はともに、32Kバイトまで拡張可能になっています。

次にVIC-1991のメモリー・マップを示します。



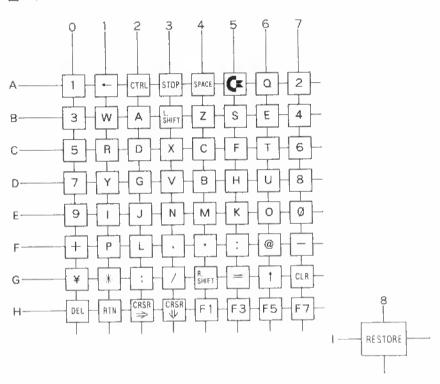


## 第6節 キーボード

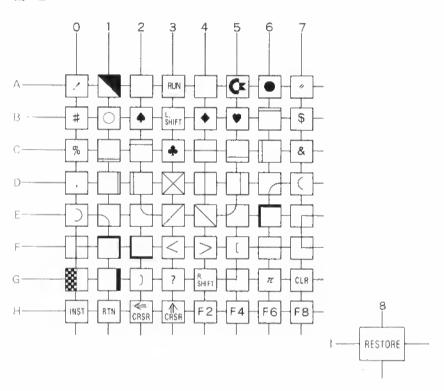
VIC-1001のキーボードは、ソフトウェア・スキャン方式をとっています。マトリックス状に配置されたキースイッチは、CPUのインプット命令によりスキャンされ、押されたキーの情報は、ソフトウェアによりコードに変換されます。

次に、キーボードマトリックスを示します。

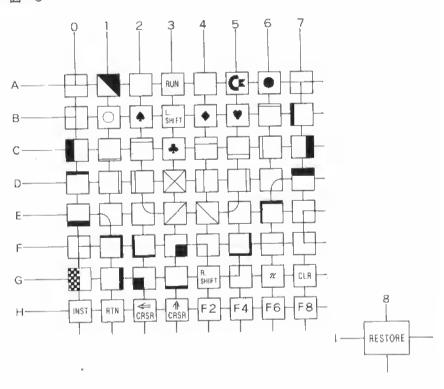
図-1



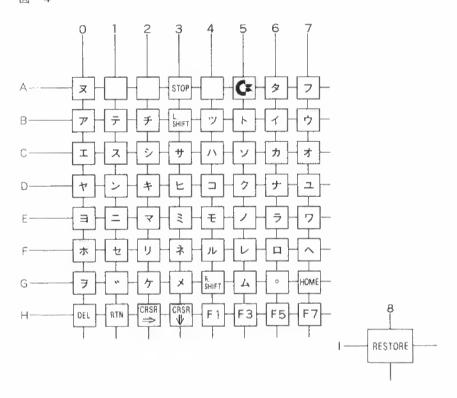


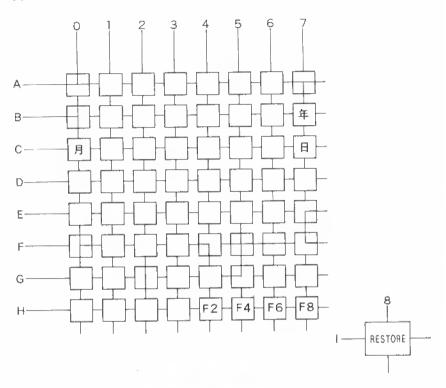












また、キーボードコネクターの端子と信号の関係は以下のようになっています。

2) KEY
3) RESTORE
4) +5V
5) COL7
6) COL6
7) COL5
8) COL4
9) COL3
10 COL2

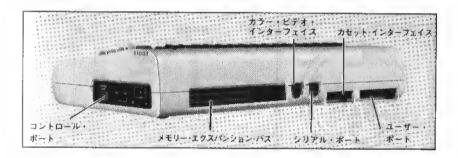
1) GND

11) COL1
12) COLØ
13) ROW7
14) ROW6
15) ROW5
16) ROW4
17) ROW3
18) ROW2
19) ROW1
2Ø) ROWØ

## 第7節 インターフェイス

VIC-1991には、以下に示すインターフェイスが標準装備されています。

- 1) ユーザー・ポート
- 2) カセット・インターフェイス
- 3) カラー・ビデオ・インターフェイス
- 4) シリアル・ポート
- 5) メモリー・エクスパンション・バス
- 6) コントロール・ポート



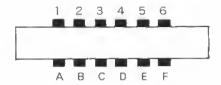
#### 1) ユーザー・ポート



ピン配列は 背面から見 た場合。以 下同

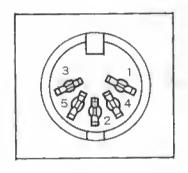
| 端子番号 | 信 号 名           | 備考         | 端子番号 | 信号名 | 備 | 考 |
|------|-----------------|------------|------|-----|---|---|
| 1    | GND             |            | А    | GND |   |   |
| 2    | +5V             | 100mA MAX. | В    | 081 |   |   |
| 3    | RESET           |            | С    | PBØ |   |   |
| 4    | JOYØ            |            | D    | PB1 |   |   |
| 5    | J0Y1            |            | E    | PB2 |   |   |
| 6    | JOA5            |            | F    | PB3 |   |   |
| 7    | LIGHT PEN       |            | Н    | PB4 |   |   |
| В    | CASSETTE SWITCH |            | J    | PB5 |   |   |
| 9    | SERIAL ATN IN   |            | K    | PB6 |   |   |
| 10   | +9V             | 100mA MAX. | L    | PB7 |   |   |
| 11   | GND             |            | M    | GB2 |   |   |
| 12   | GND             |            | N    | GND |   |   |

## 2) カセット・インターフェイス



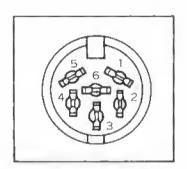
| 端子番号 | 信号名             |
|------|-----------------|
| A-1  | GND             |
| B-2  | +5V             |
| C-3  | CASSETTE MOTOR  |
| D-4  | CASSETTE READ   |
| E -5 | CASSETTE WRITE  |
| F-6  | CASSETTE SWITCH |

## 3) カラー・ビデオ・インターフェイス



| 端子番号 | 信号名        | 備    | 考   |
|------|------------|------|-----|
| 1    | +6V        | 10mA | MAX |
| 2    | GND        |      |     |
| 3    | AUDIO      |      |     |
| 4    | VIDEO LOW  |      |     |
| 5    | VIDEO HIGH |      |     |
|      |            |      |     |

## 4) シリアル・ポート



| 端子香号 | 信号                 |
|------|--------------------|
| 1    | SERIAL SRO IN      |
| 2    | GND                |
| 3    | SERIAL ATN IN/OUT  |
| 4    | SERIAL CLK IN/OUT  |
| 5    | SERIAL DATA IN/OUT |
| 6    | NC                 |

## 5) メモリー・エクスパンション・バス

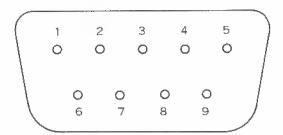
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22

A B C D E F H J K L M N P R S T U V W X Y Z

| 端子番号 | 信号名  | 備考        | 端子番号 | 信号名  | 備考            |
|------|------|-----------|------|------|---------------|
| 1    | GND  | OPUデータ    | 12   | BLK3 | ブロックセレクト3     |
| 2    | CDØ  | IN/OUT    | 13   | BLK5 | ブロックセレクト5     |
| 3    | CD1  | ÷         | 14   | RAMT | ラム・ブロックセレクト1  |
| 4    | CD2  | *         | 15   | RAM2 | 'ラム・ブロックセレクト2 |
| 5    | CD3  | #         | 16   | RAM3 | -ラム・ブロックセレクト3 |
| 6    | CD4  | "         | 17   | VR/W | VICリード/ライト    |
| 7    | 0.05 | *         | 18   | CR/W | OPUリード/ライト    |
| 8    | CD6  |           | 19   | TRQ  |               |
| 9    | CD7  |           | 20   | NO   | ノーコネクション      |
| 1.0  | BLK1 | ブロックセレクト1 | 21   | +5∀  | MAX 5ØØmA     |
| 11   | BLK2 | ブロックセレクト2 | 22   | GND  |               |
|      |      |           |      |      |               |

| <b>岩子番号</b> | 信号名  | 備 考        | 端子番号 | 信号名   | 備考          |
|-------------|------|------------|------|-------|-------------|
| A           | GND  |            | N    | CAIØ  | OPUアドレス10   |
| В           | CAØ. | CPUアドレスの   | P    | CAII  | <i>∗</i> 11 |
| C           | CA1  | * 1        | R    | OA12  | · 12        |
| D           | CA2  | <i>*</i> 2 | S    | CA13  | v 13        |
| E           | CA3  | // 3       | T    | 1/02  | 1/0セレクト2    |
| F           | CA4  | ø 4        | U    | 1/0/3 | » 3         |
| Н           | CA5  | ø 5        | V    | 502   | システム・クロック2  |
| J           | CA6  | » 6        | W    | NMI   |             |
| K           | CA7  | ÷ 7        | X    | RESET |             |
| L           | CAB  | · 8        | Y    | NC    | ノーコネクション    |
| M           | CA9  | v 9        | Z    | GND   |             |

## 6) コントロール・ポート



| 端子番号 | 信号名       | 備考         |
|------|-----------|------------|
| 1    | JOYØ      |            |
| 2    | JOY1      |            |
| 3    | JOY2      |            |
| 4    | JOY3      |            |
| 5    | POT Y     |            |
| 6    | LIGHT PEN |            |
| 7    | +5V       | MAX. 100mA |
| 8    | GND       |            |
| 9    | POT X     |            |

## 第8節 CRT制御方式

VIC-1661では、画面に文字および図形を表示するために専用のビデオ・インターフェイス・チップ (MPS-6566) を使用しています。

画面に表示される文字および図形のデータは、RAM上に配置されたビデオRA M上に格納されます。DMAコントローラは、このビデオRAM内のデータをD MA転送(ダイレクト・メモリ・アクセス)により、文字および図形データをそのバッファに一時記憶し、画面に同期してビデオ信号発生回路に出力します。

ビデオ信号発生回路は、タイミング回路によって発生された同期信号と、映像 信号とを合成して、コンポジットビデオ信号を発生させます。

#### ビデオRAM

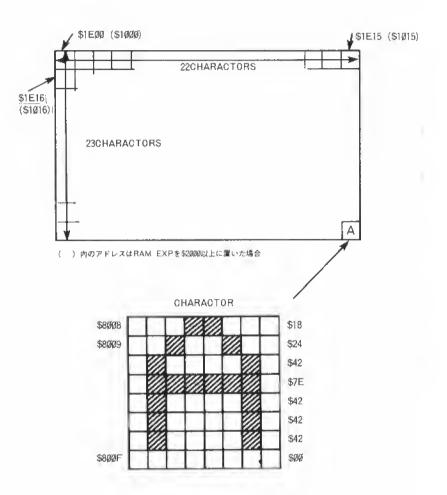
VIC-19901のビデオRAMは、メイン・メモリーのアドレス\$1E990~1FFFの512  $\cap$  イトに配置されています(ただし、\$2009以上にRAMが増設されている場合には、\$10090~11FFとなります)。

次ページに画面とビデオRAMの対応を示します。

#### BASICテキストファイル

VIC-1 $\emptyset$ 01のBASICテキスト・ファイルは、通常 \$ 1 $\emptyset$ 90番地からスタートします。 ただし、\$ $\emptyset$ 4 $\emptyset$ 9~ $\emptyset$ FFFにRAMが増設されている場合には、\$ $\emptyset$ 4 $\emptyset$ 9がスタート番 地になります。しかし \$ 2 $\emptyset$ 9 $\emptyset$ 0以上にもRAMが増設されている場合は \$1 $\emptyset$ 9 $\emptyset$ 8番地か らスタートし、\$ $\emptyset$ 4 $\emptyset$ 9 $\emptyset$ 9 $\emptyset$ FFFはテキスト・ファイルとしては扱われません。

#### ビデオRAMの構成



# 第2部 ● ソフトウェア編

# 第1章 CBM BASICの概要

CBM BASIC (V2)は、VIC-1M1のプログラミング言語です。パーソナル・コンピューターのパイオニアであるPET/CBMのコモドールBASICに基づいており、コモドールBASIC同様、数多くあるBASIC言語の中でも、特に分りやすく、使いやすい言語です。

この章では、CBM BASICの概要を説明し、第2章でコマンド・ステートメントの、第3章で関数の説明をおこないます。第4章では、CBM BASIC を使ったプログラム例をいくつかあげます。

## 第1節:初期状態

コールドスタートするためには、次の3通りの方法があります。

①電源の投入

システムのリセット、およびOS (オペレーションシステム) とBASIC の変数を すべて初期状態にします。

- ②リセット信号 メモリー拡張用コネクターよりリセット信号を送ります。
- ③コマンドによるリセット SYSまたは、ブランチ命令により、リセット番地へジャンプします。 以上、いずれの場合も、ディスプレイ上は、次のように表示されます。

\*\*\*\* CBM BASIC V2 \*\*\*\*

3583 BYTES FREE

READY.

## 第2節:動作モード

VIC-1001に電源を入れ、BASICが走るようにすると、ディスプレイ上で、カーソルが点滅しています。これは、BASICが、コマンドを受け取る準備のできていることを意味します。BASICは、ダイレクト・モードか、または、プログラム・モードのどちらかで使用することができます。

7イレクト・モード……この場合、BASICのステートメントおよびコマンドにライン・ナンバーをつけません。また命令が人力された時点で、すぐに実行されます。算術および論理演算の結果を直ちに表示したり、後で使うために格納することができますが、命令は実行後失なわれます。このモードは、デバッグをおこなう場合や、BASICを完全なプログラムを必要としない簡単な計算をすぐにおこなうカリキュレーターとして使うのに便利です。

プログラム・モード……このモードはプログラムを入力する場合に使用します。 プログラムの各ライン(行)は、ライン・ナンパーでは じまり、メモリーに格納されます。メモリーに格納され たプログラムはRUNコマンドを入力することにより実 行されます。

## 第3節:ラインの書式

ラインはBASICプログラムの基本的な単位です。BASICのラインの書式は次かとおりです。

n n n n n n BASICステートメント〔:BASICステートメント……〕〈キャリッジ <sub>ライン・ナン</sub>に リターン〉

プログラムは1ラインに1つ以上のBASICステートメントを入れることができ 上すが、各々のステートメントはコロン(:)で区切られていなくてはなりません。 BASICプログラムは必ずライン・ナンバーではじまり、キャリッジリターンで 終ります。1ラインにつき最大88文字まで入れることができます。

## 第4節:ライン・ナンバー

BASICプログラムは必ずライン・ナンバーではじまります。ライン・ナンバーはプログラムがメモリーに格納される順番をあらわしています。また、プログラム中での分岐や編集をおこなう場合の目印として使用されます。

ライン・ナンバーは Øから63999までの整数を使います。

## 第5節:キャラクターセット

CBM BASICのキャラクターセット(使用文字)は、英文字、数字、グラフィック文字、カナ文字、そして特殊文字があります。特殊文字および特殊キーは次の通りです。

```
特殊文字
        名称
        プランク (空白)
        セミコロン
        イコール、等号
 -
 +
        プラス、加算の演算子
        マイナス、減算の演算子
        アスタリスク、乗算の演算子
 *
        スラッシュ 、除算の演算子
 †
        上向き矢印、べき乗の演算子
 (
        左かっこ
        右かっこ
        パーセント
%
 #
        ナンバー記号
 $
        ドル記号
 Ţ
        感嘆符
        左かぎかっこ
 ]
        右かぎかっこ
        コンマ
        終止符(ピリオド)または小数点
        シングルクォーテーション(アポロストロフィ)
        ダブルクォーテーション(引用符)
        コロン
        アンド
 &
 ?
        疑問符 (クエスチョン・マーク)
        不等号(より小さい)
        不等号(より大きい)
 (0)
        アット・マーク(単価記号)
        左向き矢印
 ¥
        田記譽
        円周率(3.14159265)
```

特殊キー 機能

《CLR》 ディスプレイ画面を消去して、カーソルをホームポジ

ション(左上スミ)に戻します。

〈CRSR⇒〉 カーソルを右へ1文字ずらします。

〈CRSR⇔〉 カーソルを左へ1文字ずらします。 〈CRSR介〉 カーソルを上へ1文字ずらします。

〈CRSR⊕〉 カーソルを下へ1文字ずらします。

《CTRL》 数字の1から8と併用することにより、ディスプレイ

画面の色を変えることができます。また、CTRL+ Rで、文字を反転させることができ、CTRL+ Øで反 転を解除できます。さらにまた、このキーを押すこと

により、スクロール速度を落とすことができます。

〈DEL〉 カーソルの左側にある文字を1字削除し(それより右

の文字を左へ詰め)ます。リピート機能付き。

〈HOME〉 カーソルをホームポジション(左上スミ)に戻します。

〈INST〉 カーソルの位置とその左側の文字との間にスペースを

入れ、文字の挿入を可能にします。

《RESTORE》 STOP キーと同時に押すことにより、システムのリ

セットをおこなわずに、BASICを再スタートできます。

〈RETURN〉 1行の入力を終了させます。そのさい、カーソルを次

列の左端に移します。カーソルが最下段にある場面は、

画面をスクロールします。

〈RUN〉 LOAD+RUNのコマンドを入力したのと同じ働きをし

ます。

〈SHIFT〉 キャラクター・セットの変換に用います。

〈SHIFT LOCK〉 SHIFT状態のキャラクター、セットに保ちます。

〈STOP〉 プログラムの実行を中止して、BASICのダイレクト・

モードに戻します(ディスプレイにはREADY, が表示 され、カーソルが点滅します)。また、プログラムをカ セット・ドライブからLOADするのを中止させる時に

も用います。

SHIFT キーと同様の働きをします。 SHIFT キー

と併用して、グラフィック・キャラクター・モード◆→ 片仮名キャラクター・モードの変換をおこないます。

また、片仮名モード [と][の切換えをおこないます。

# 第6節:定数

定数はBASICが実行時にそのまま使用する値です。定数には文字定数および数値定数の2種類があります。

文字定数は引用符(ダブルクォーテーション)に囲まれた最大255文字までの文字の列です。文字定数の例をあげましょう。

"COMMODORE"

"1980"

″コモドール"

数値定数は正または負の数です。BASIC中での数値定数は ゙, ″(コンマ) を持つことができません。

数値定数には次の2つの形式があります。

### 1. 整数形式

-32768から+32767までのすべての整数。整数は小数点をふくむことはありません。

# 2. 浮動小数点形式

指数形式で表現された正または負の数値です。浮動小数点形式では数値(仮数部)に続き文字Eそして符号(+のばあいはなくてもよい)つきの整数(指数部で表現されます。指数部の範囲は-39から+38までです(CBM BASICで扱かえる実数は、 $2.93873588E-39\sim1.70141183E+38$ までです)。CBM BASICでは、浮動小数点形式の数値定数は、10桁の精度で格納され、9 桁までの桁数で表示されます。

浮動小数点形式の数値は、次のいずれかに該当します。

- 1.10桁以下の数値
- 2. 指数形式で表示
- 3. 浮動小数点形式の変数として割りあてられた場合、

# 例) 28.9

-4.09E-05

# 第7節:変数

変数とはBASICプログラム中で使われる値をあらわすために用いられる名前です。変数の値はプログラマーにより定義されたり、プログラムによる演算の結果が割りあてられたりします。変数は値が割りあてられるまでは  $\S B''$  (ゼロ)とみなされます。

# 1変数名および型宣言文字

BASICでは変数名は何文字でもかまいませんが、最初の2文字だけが意味をも ち、それにより区別されます。最初の1文字は必ず英文字でなければなりません が、残りの文字は英文字または数字または型宣言文字のどれでもかまいません。

変数名は予約語(BASICで使用される、コマンド・ステートメント・関数名な ついてはなりません。また予約語を含んでいてもいけません。たとえばBG 〇という変数名はG 〇という予約語を含んでいるので認められません。

変数は数値または文字列のいずれもあらわすことができます。文字変数名は最度に **\$** (ドル記号)をつけて表現します。

例) A\$ = "ウリアゲニッポウ" AB\$ = "ABC"

この \$ (ドル記号)は、変数が文字列であることを宣言する型宣言文字です。 数値変数が、整数変数であることを宣言するのには変数名の最後に <sup>%</sup>%"(パー ピント記号) をつけておこないます。

それら以外の変数名を用いる場合は、浮動小数点形式の変数になります。

#### 例)(変数名)

MI 浮動小数点形式の変数

 LIMIT%
 整数変数

 N\$
 文字変数

#### 2 配列変数

配列は、同じ変数名で参照することのできる値の集まり、またはテーブルです。 配列のそれぞれの要素は、整数または整数表記による添字を伴った配列変数により参照されます。配列変数名はその配列の次元の数と同じ個数の添字を持ちます。 たとえば、V(10)は1次元配列の値として、D(1.4)は2次元配列の値としてあつかわれます。2次元以上の場合も同様です。

# 第8節:型の変換

BASICは、必要に応じて数値定数の型を、ある型から他の型に自動的に変換します。この場合、以下に述べる点について留意して下さい。

1. ある型の数値変数が他の異なった型の数値変数に割りあてられた場合は、その変数名により宣言された型に変換して格納されます。(もし文字変数が数値に、またはその逆が割りあてられた場合は、TYPE MISMATCH"エラーが発生します。)

例) 10 A %=12.34 29 PRINT A % RUN 12

- 2. 式を計算する場合、算術または論理演算のオペランドはすべて浮動小数点形式の同じ精度でそろえられます。整数は評価のためにいったん浮動小数点形式に変換され、再度整数形式に戻されます。
- 3. 論理演算では、オペランドは整数に変換され、整数の結果が得られます。オペランドが-32768から+32767の範囲内でない場合は  $^{\circ}$ OVER FLOW"エラーが発生します。
- 4. 浮動小数点形式で表わされた数値が整数に変換される場合には、小数部分は 丸められ、変換された結果の整数はもとの浮動小数点形式の数値より小さい か等しい値をもちます。

例) 10 AZ=12.24 20 PRINT A% RUN 12

# 第9節:式と演算

式とは単に文字または数値定数、または変数、あるいはある値を得るために定 数や変数を演算子で結合したものです。

演算子は与えられた値について、算術または論理演算を実行します。BASICに 備わっている演算は大きくわけて次の4つに分類することができます。

- 1. 算術演算
- 2. 関係演算
- 3. 論理演算
- 4. 関数

# 门算術演算

算術演算子を演算の順位にしたがって、ならべると次のようになります。

| (演算子) | (演算)        | (例)          |
|-------|-------------|--------------|
| †     | 指数          | X † Y        |
| _     | 否定(負号)      | - X          |
| *,/   | 乗算、浮動小数点の除算 | X * Y, X / Y |
| +, -  | 加算、減算       | X+Y, $X-Y$   |

演算の順序を変える場合は、カッコ \*( \*," )" を使用します。カッコ内の演算 一最初におこなわれます。カッコの内部では通常の演算順位が適用されます。

演算をおこなう場合、カッコのネスティング(入れ子)使用は最大10個までで

通常の数式表現とそれらをBASICで表現した例を示しましょう。

| 数式表現                | BASICでの表記                        |
|---------------------|----------------------------------|
| $\times$ + 2 Y      | X + 2 * Y                        |
| $X - Y \div Z$      | X — Y / Z                        |
| $X \times Y \div Z$ | X * Y / Z                        |
| (X*) Y              | (X † 2) † Y                      |
| X (Y*)              | X † (Y † 2)                      |
| X ( Y)              | X * (一 Y)演算子が連続する場合はカッコで区切って下さい。 |

#### ●注意●

Øによる除算とオーバーフロー:数式にたいする演算処理の実行中に除数が Ø の場合には、\*DIVISION BY ZERO" エラーが発生します。また、オーバーフローが生じた場合には、\*OVER FLOW" エラーが発生します。

## (2)関係演算子

関係演算子は2つの値を比較するのに用いられます。比較の結果は〝真″(-1)、または〝偽″ $(\emptyset)$  になります。この結果を用いてプログラムの流れの決定したりすることができます(2-13を参照)。

| 演算子 | 説明      | 表記        |
|-----|---------|-----------|
| =   | 等しい     | X = Y     |
| < > | 等しくない   | X <> Y    |
| <   | より小さい   | x < y     |
| >   | より大きい   | $x{>}y$   |
| <=  | 等しいか小さい | $X \le Y$ |
| >=  | 等しいか大きい | X >= Y    |

(符号は変数に値を代入するのにも使われます。2-16 LETを参照)

1つの式の中で関係演算子と算術演算子が同時に使われている場合、算術演算 子が先に実行されます。

たとえば

$$X+Y < (T-1)/Z$$

この場合、X+Yの値が  $(T-1)\div Z$ の値より小さければ真となります。関係演算子の使用例としては、次のような例があります。

### (3)論理演算子

ビット操作や、論理演算をおこなったり、いくつもの関係を調べたりするため論 理演算子を使用します。論理演算子はそのオペランドの値により、真または偽の 値を結果として与えます。1つの式の中では、論理演算は算術および関係演算のあ とで実行されます。論理演算の結果を次の表に示します。各演算子は演算の順位 にしたがってリストされています。

| NOT (否定) |       |         |
|----------|-------|---------|
| X        | NOT X |         |
| 1        | Ø     |         |
| Ø        | 1     |         |
| AND(論理積) |       |         |
| X        | Υ     | X AND Y |
| 1        | 1     | 1       |
| 1        | Ø     | Ø       |
| Ø        | 1     | Ø       |
| Ø        | Ø     | ø       |
| OR (論理和) |       |         |
| X        | Υ     | X OR Y  |
| 1        | 1     | 1       |
| 1        | ø     | 1       |
| Ø        | 1     | 1       |
| Ø        | ø     | ø       |
|          |       |         |

関係演算子がプログラムの流れを決定するのに使用できるのと同様に、論理演算子も2つまたは、それ以上の関係を結びつけて、判断のための真または偽の値を与えることができます(2-13 IF を参照)。

# 例) IF B >200 AND F <5 THEN 100

IF I<10 OR J<0 THEN50

IF NOTP THEN 200 (Pが-1以外はすべてライン・ナンバー200ヘジャンプします)

論理演算子は、オペランドを一32768から、+32767までの16ビットの符号付き、2の補数表示の整数に変換してから演算をおこないます(もしオペランドがこの範囲になければエラーが発生します)。

指定された演算はこの整数に対し、ビットごとに対応させておこなわれます。つ より、結果の各ビットは2つのオペランドの対応するビットの状態によって決定さ れます。したがって、パイト単位のデーターを、あるビットパターンに対して調 べるのに論理演算を使うことができます。

たとえば、あるL/Oポートのステータスパイトのある1ビットだけをマスクする のに、AND演算子を使うことができます。またOR演算子は、2つのパイトデータを併合して1つの2進数を得ることができます。

次の例は、論理演算子の働きを理解するのに役に立つでしょう。

NOTX=-(X + 1) 任意の数の 2 の補数は各ビットを反転させたものに 1 を加えたものです。

#### 4 関数

関数は式の中で、あるオペランドに対して決められた演算操作を呼びだすのに使用されます。CBM BASICにはSQR (n-1)やSIN (サイン) などの組み込み関数が用意されています。組み込み関数については第3章で説明します。CBM BASICではまた、ユーザーにより、関数を定義することができます  $(2\cdot600)$  EF FNを参照)。

# (5)文字列の演算

文字列は"+"によってつなぐことができます。たとえば

IN 18 A\$="VIC": B\$="1001"

28 PRINT AS+BS

30 PRINT "COMMODORE" + A\$+B\$

RUN

VIC1001

COMMODOREVICI 001

+た、数値演算に用いたのと同じ次の比較演算子によって比較することもできょ 4

# <> < > <= >=

文字列の比較は1度にそれぞれの文字列から1文字ずつ、その ASCII(アストー)コードの比較をおこないます。もしすべてのASCII(アスキー)コードが等しければ、その2つの文字列は等しくなります。もしASCII(アスキー)コートが異なった場合は、小さいコードのものが数値的に小さいとされます。もし比利の途中で一方の文字列が終わりになった場合は、その短い文字列の方が小さいされます。文字列の前と後ろのスペース(空白)も意味をもちます。

|刑) "AA"<"AB"
"COMMODORE"="COMMODORE"
"X\$">"X#"
"CL ">"CL"
"VIC"<"VIC1991"
A\$="8/16/86" の場合
A\$<"9/10/86"

# 第9節 スクリーン・エディター

第1部ハードウェア編を参照して下さい。

# 第10節 エラーメッセージ

BASICプログラムを実行中に何か処理を中断させるエラーが検出された場合、エ ラーメッセージが表示されます(付録Cにエラーメッセージの一覧表があります)。

# 第2章 CBM BASICの コマンドとステートメント

この章ではCBM BASICの持つすべてのコマンドとステートメントについて、 以下の形式で説明します。

書式: 命令の正しい書式 (フォーマット) を示します。使用される記号の意味

については下記を参照して下さい。

目的: なんのためにその命令が使用されるのかを説明します。

説明: 命令の使用法を詳しく説明します。

例: 命令の使用法を示すプログラム例をあげます。

# 書式の記法

各コマンド・ステートメントの書式は、以下の記法にしたがっています。

- 1. アルファベットの大文字で記された項目は、そのままの形、または省略形で 入力して下さい(省略形については付録Bを参照して下さい)。
- 2. 山カッコ (〈 〉) で、囲まれた項目はユーザーが指定する項目です。
- 3. かぎカッコ (〔〕) で、囲まれた項目は、オプションです。
- 4. 山カッコおよびかぎカッコ以外のコンマ、丸カッコ、セミコロン、ハイフォン、等号などの記号は、示された位置に正しく入力して下さい。
- 5. 省略記号…の続く項目は、一行の許す長さ内で任意の回数繰り返すことができます。
- 6. たて線(1)で区切られている項目は、各々のうち1個を選択します。

## CBM BASICコマンド・ステートメント一覧表

| CLOSE                          | 2. 1 | LIST              | 2.17 |
|--------------------------------|------|-------------------|------|
| CLR                            | 2. 2 | LOAD              | 2.18 |
| CMD                            | 2. 3 | NEW               | 2.19 |
| CONT                           | 2. 4 | ON~GOSUB, ON~GOTO | 2.20 |
| DATA                           | 2. 5 | OPEN              | 2.21 |
| DEF FN                         | 2.6  | POKE              | 2.22 |
| DIM                            | 2. 7 | PRINT, PRINT#     | 2.23 |
| END                            | 2. 8 | READ              | 2.24 |
| FOR~NEXT                       | 2.9  | REM               | 2.25 |
| GET, GET#                      | 2.10 | RESTORE           | 2.26 |
| GOSUB~RETURN                   | 2.11 | RUN               | 2.27 |
| GOTO                           | 2.12 | SAVE              | 2.28 |
| IF $\sim$ THEN. IF $\sim$ GOTO | 2.13 | STOP              | 2.29 |
| INPUT                          | 2.14 | SYS               | 2.30 |
| INPUT#                         | 2.15 | VERIFY            | 2.31 |
| LET                            | 2.16 | WAIT              | 2.32 |

## 2.1 CLOSE

書式: CLOSE(ロジカルファイル番号)

目的: ファイルに対する人出力を終了する。

説明: 〈ロジカルファイル番号〉は、あるファイルがオープンされた時に使われた番号です。CLOSEが実行されますと、特定のファイルとファイル番号の間のつながりはなくなります。そしてそのファイルは、異なったファイル番号ないしは同じ番号で再度オープンすることができます。同様に、一度使用されたファイル番号を他のファイルに用いることもできます。

シーケンシャル出力ファイル作成時にCLOSEを実行しますと、最後の出力パッファを書きこみます。

例: OPEN 4, 4 PRINT#4, "THIS GOES TO PRINTER" CLOSE4

## 2.2 CLR

書式: CLR

目的: すべての数値変数を  $\emptyset$  に、文字変数をヌルコード  $((\emptyset\emptyset)_{16})$ にセットし、

メモリーのエンド、スタック、配列をリセットします。

談明: CLRがBASICプログラム中で実行された後、次に述べる以外はプログラ ムを続行することができます。

●エラーになり続行できない場合

- ●スタックを使用する処理を実行しようとした場合(GOSUB~RETU RN. FOR~NEXTなど)
- ●配列を使用する場合、つまりDIMで配列を宣言した後、CLRを実行す ると、その後では配列は宣言されていないものとして扱われます。

例 10 X=25

20 CLR

30 PRINT X

RUN

Ø

#### 2.3 CMD

**書式:** CMD (ロジカルファイル番号) [. 印字する数値または文字]

目的: シリアルバスと機器をつなぎ、その機器をリスナー状態にします。

説明: CMDはPRINT#(2,23参照)と同じパラメータを用います。

REM CREATE HARDCOPY LISTING 例

OPEN 4. 4

CMD 4, "PROGRAM LISTING"

LIST

PRINT#4. "TURNS OFF CMD"

CLOSE 4

#### 2.4 CONT

書式: CONT

目的: ストップキー入力後、またはSTOP (2.29参照) やEND (2.8参照)を

実行した後に、プログラムを再実行する。

説明: 処理を中止した個所から実行が継続されます。INPUT (2.14参照) ステートメントからプロンプト出力後に処理を中止した場合、CONTを実行すると、再びプロンプトを出力するところから実行を開始します。通常 CONTは、デバッグのためにSTOPと共に用いられます。プログラムの実行を途中で中止し、ダイレクトモードステートメントにより、変数などの中間結果を調べたり変更することができます。そして、その後、CONTやGOTO (2.12参照) を用いて処理を再開することができます。GOTOの場合は、処理再開するスタートのプログラム・ライン・ナンバーを指定することができます。

実行の中断中に、プログラムが変更ないし編集された場合は、CONT は無効になります。また中断中、ダイレクトモード実行中にエラーが発生した場合は処理を続行することができません。

プログラムの実行がエラーにより中断された場合はCONTにより実行を再開することはできません。

例 : 2.29 STOPの項を参照

## 2.5 **DATA**

書式: DATA⟨定数⟩[,⟨定数⟩…]

目的: プログラム中のREAD (2.24参照) により読まれる数値および文字定数 を格納する。

説明: DATAステートメントは、非実行ステートメントで、プログラム中どこに置いてもかまいません。DATAステートメントは、(コンマに区切られて) 1行中にはいるだけの定数をいれることができ、またプログラム中にいくつでもDATAステートメントを使うことができます。READステートメントは、ライン・ナンバーの順にDATAステートメントを読んでゆきます。そしてそこに含まれているデータはその行にいくつデータがあるのか、またはその行がプログラム中のどこに置かれているかにはかかわらず、1つの連続したデータの一部と考えることができます。

〈定数〉は任意の形式の数値定数、すなわち間定小数点、浮動小数点、 または整数を含むことができます(数式は許されません)。DATAステートメントに含まれる文字定数は、その中にコンマ(,)、コロン(:)または前後に意味のある空白(スペース)を含む場合にはクォーテーション(")で定数を囲んで下さい。それ以外の場合には、その必要はありません。 READステートメントにより、指定される変数の型(数値または文字)は対応するDATA文の定数と一致しなくてはなりません。

DATAステートメントは、RESTORE (2.26参照) により最初から読むことができます。

例 : 2.24 READの項を参照して下さい。

#### 2.6 DEF EN

**書式**: DEF FN (名前) [((パラメータ))] = (関数の定義)

目的: ユーザーによって書かれた関数を定義し名前を付ける。

説明: 〈名前〉は正しい変数名でなければなりません。FNに引き続く〈名前〉が関数の名前になります。〈パラメータ〉は、その関数が呼ばれた時に置き換えられる、関数の定義式の中の変数名です。〈関数の定義〉は、その関数の演算内容を記述する式で、1行の範囲にかぎられます。この式にあらわれる変数名は、単に関数を定義するだけで、同じ名前をもつプログラム中の変数には何の影響も及ぼしません。また関数の定義式に使われた変数名はパラメータ中にあってもなくてもかまいません。もしあれば、そのパラメータの値は関数が呼ばれた時に与えられます。そうでなければ、その変数の現在の値が使われます。

このユーザ定義関数は、数値関数のみであって文字関数は許されていません。関数名によって型が指定されたなら、式の値は、その型として扱かわれます。もし関数名により指定された型と引数(アーギュメント)の型が一致しない場合は \*TYPE MISMATCH" エラーが発生します。

DEF FNステートメントはそれが定義する関数が呼ばれる前に実行されなければなりません。もし定義される前に関数が呼ばれた場合は、 \*UNDEF'D FUNCTION"エラーが発生します。

なお、DEF FNはダイレクトモードでは使用できません。

例: 10 DEF FNAB(X)=X 1 3 / Y 1 2

20 1 = 2 : Y = 3

30 T=FNAB(I)

40 PRINT T

RUN

.88888889

ライン・ナンバー10で関数FNABを定義します。

ライン・ナンバー30では、10で定義した関数が呼ばれます。

#### 2.7 DIM

書式: DIM(変数名)((添字の最大値))

目的: 配列変数の添字の最大値を指定し、同時にメモリー領域を割り当てる。

説明: DIMステートメントなしに配列変数名を用いた場合、その添字の最大値は、10と見なされます。もし指定された値より大きい添字が使われた場合は、BAD SUBSCRIPT″エラーが発生します。添字の最小値はつねにのです。

DIMステートメントは指定された配列のすべての要素の値を初期値 Ø に設定します。

1次元以上、255次元までの配列を宣言することができます。添字の最大値は32767まで使用することができます。配列全体の大きさは、使用できる、メモリーの容量により制限されます。

例: 10 DIM A (20)

20 FOR I=0TO20

30 READA(I)

40 NEXT

50 DATA1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21

#### 多次元配列の場合

10 DIM R3 (5 . 5)

10 DIM D\$ (2,3,2)

#### 2.8 END

書式: END

目的: プログラムの実行を終了させ、すべてのファイルを閉じ(クローズ)、 その後、コマンドレベルに戻る。

説明: ENDステートメントはプログラムの実行を終了させるために、プログラム中のどこに置いてもかまいません。

STÖP (2.29参照)とは異なり、ENDステートメントではBREAKメッセージは出力されません。また、プログラムの一番最後のENDはなくてもかまいません。ENDステートメントを実行すると BASIC はつねにコマンドレベルに戻ります。

例 : 100 IF K>1000 THEN END

#### 2.9 FOR~NEXT

書式:  $FOR\langle g数名\rangle = \langle X \rangle TO\langle Y \rangle [STEP\langle Z \rangle]$ 

NEXT [〈変数名〉] [,〈変数名〉…] ただしX, YおよびZは数値式

目的: 一連の命令を指定した回数分、繰り返し実行する。

説明: 〈変数名〉で指定した変数は繰り返しの回数をかぞえるためのカウンターとして使用されます。最初の数値式〈X〉は、カウンターの初期値、二番目の数値式〈Y〉はカウンターの最終値を設定します。次にFORステートメントからNEXTステートメントまでのプログラム文が実行されます。そしてNEXTをみつけた時点でカウンターにはSTEPによって指定された値だけ加算されます。もしSTEPが省略された場合は、+1 加算されます。そしてカウンターの値が最終値より大きいかどうか調べられ、もし大きくないならBASICは、FORの次のステートメントに戻ります。もし大きいならば、NEXTに続くプログラム文が実行されます。

STEPで指定する値が負の場合は、カウンターの初期値は最終値より大きい値を設定して下さい。この場合カウンターの値が初期値より小さくなるまで、FORステートメントからNEXTステートメントまでのプログラム文の実行が繰り返されます。

# 多重ループについて(ネスティング)

FOR...NEXTループは入れ子構造(ネスト)にすることができます。つまり、FOR...NEXTループの中に、他のFOR...NEXTループを置くことができます。ループが多重になった場合は、それぞれのループのカウンターには、各々異なった独自の変数名を使用しなくてはなりません。内側のループのNEXTステートメントは、外側のループのものより以前にあらわれなければなりません。多重ルーブが同じ場所で終わる場合は、それら全部のループに対して、1つのNEXTステートメントに、カウンター変数をループの順番に、コンマで区切って書くことで、対応することができます。NEXTステートメントが、最近のFORステートメントに対応する場合には、NEXTの変数名は省略することができます。もし対応するFORステートメントのないNEXTがあらわれたなら、NEXT WITHOUT FOR"エラーが発生し、プログラムの実行は停止します。

スタックの容量に制限があるため、多重ループで使用できる FOR... NEXTの数は最大9個までです。

```
例 1 : 多面ループ
       10 FOR I= 1 TO 3
       20 FOR J=1TO3
       30 PRINT I: J
       40 NEXT J.I
       RUN
         1
            1
         1
         1
            3
         2
           - 1
         2
           2
         2
           3
         3
           - 1
         3 2
州 2 : ループカウンター設定後、使用した変数名に別の値を代入
       10 K=10
       20 FOR I=1TO K STEP 2(カウンターは1~10に設定)
       30 PRINT I:
       40 K=K+10
       50 PRINT K
       60 NEXT
       RUN
         1
            20
         3 30
         5 40
         7 50
```

M 3 : カウンターの最終値が初期値より小さい場合

```
10 J=0
20 FOR I=1TO J
30 PRINT I
40 NEXT I
RUN
1
```

9 60

この場合、カウンターが、最終値(J)より大きいということは、NEXTが実行された段階で、はじめてわかるため、ループが1回だけ実行されます。

例4: 10 FOR I=1TO I+5

20 PRINT I;

30 NEXT

RUN

1 2 3 4 5 6

この場合、ループ回数は6回です。ループカウンターの初期値が最終値 より先に変数名にセットされます。

注:整数型変数 (1%、A%のように変数名の後に%をつけてあらわします) を、ループのカウンター変数として使用することはできません。

10 FOR 1%= 1 TO10

20 PRINT 1%

30 NEXT 1%

RUN

? SYNTAX ERROR IN 10

2.10 GET, GET#

**書式:** GET[# (ロジカルファイル番号),](変数名)

目的: ファイルから1文字読みこんで、変数に代入する。

説明: ロジカルファイル番号をともなわないGETは、キーボードバッファー から、数字でも文字でも、1文字だけをバッファー内から入力します。 もしバッファー内に何もなければ、ヌルコード (=(ØØ) 16)、値はØを入力します。

GET # は、ロジカルファイル番号により指定されたファイルより 1 文 字入力します。ただし、 $\overline{O}$ PENステートメント(2.21参照)により指定されたデバイス番号(機器番号)が  $\emptyset$  の場合は、前述の  $\overline{G}$ E T と同じ動作をします。デバイス番号が 1 (= カセット) の場合  $\overline{E}$ OF (= エンドオブファイル)のキャリッジリターン(= ( $\emptyset$ D) 16)を入力したかどうか、ST (ステータス)を測べることによって検出することができます。

例: 10 PRINT"HANGS UNTIL KEY PRESSED"

20 GET AS

30 IF A\$=""THEN20

40 PRINT "OK"

# 2.11 GOSUB~RETURN

書式: GOSUB 〈ライン・ナンバー〉

- 1

RETURN

目的: サブルーチンへの分岐および戻り。

説明: 〈ライン・ナンバー〉は、サブルーチンの最初の行のライン・ナンバーです。

サブルーチンのRETURNを実行すると、BASICは最近のGOSUBステートメントの次のステートメントに戻ります。1つのサブルーチン中にRETURNがいくつあってもかまいませんが、プログラムのロジックは、サブルーチン中の正しいRETURNステートメントを指示するようになっていなくてはいけません。サブルーチンは、プログラムのどこに置かれてもかまいませんが、メインプログラムと、一目で区別できるように書かれることをおすすめします。また不用意にサブルーチンに飛びこんでしまうことを防止するためには、STOP、ENDないしはサブルーチンを飛びこすよう指示するGOTOステートメントをサブルーチンの前に置くようにして下さい。

サブルーチンを、プログラムの先頭にもってきて、小さいライン・ナンバーを与えることにより、実行速度をたかめることができます。

スタックの容量に制限があるため、GOSUB~RETURNをネスティング (2.9 FOR~NEXT参照) で使用した場合、ネスティングできるのは最大23個です。

- 例 : 10 GOSUB 40
  - 20 PRINT"END"
  - 30 END
  - 40 PRINT"SUBROUTINE":
  - 50 PRINT"SUBROUTINE":
  - 60 PRINT"SUBROUTINE"
  - 70 RETURN

RUN

SUBROUTINE SUBROUTINE

END

#### 2.12 GOTO

書式: GOTO 〈ライン・ナンバー〉

目的: 無条件に、通常のプログラムの流れから指定された行へ分岐する。

説明: 指定された〈ライン・ナンバー〉の行にあるステートメントが、実行文であれば、その部分およびそれに引き続くステートメントを実行します。もし非実行文であれば、〈ライン・ナンバー〉の行の後の最初の実行文より実行が継続されます。

例: 19 READ R

20 PRINT "R=";R,

30 A=3.14 × R 1 2

49 PRINT "AREA=":A

59 GOTO10

60 DATA5,1,15

RUN

R = 5

AREA= 78.5

R = 1

AREA = 3.14

R= 15

AREA= 706.5

?OUT OF DATA ERROR IN 10

#### 2.13 IF~THEN, IF~GOTO

書式: 1F〈論理式〉THEN〈ステートメント〉|〈ライン・ナンバー〉

IF (論理式)GOTO (ライン・ナンバー)

**目的**: 〈論理式〉によって記述される条件にしたがって、プログラムの実行の 流れに関する判断をおこなう。

説明: 〈論理式〉の結果が真  $(\neq\emptyset)$  ならばTHENまたはGOTO文が実行されます。THENには、分岐するライン・ナンバーまたは実行するステートメントのどちらも続けることができます。

もし〈論理式〉の結果が、偽(= Ø)ならば、THENまたはGOTO文は無視され、次の行のステートメントの実行がおこなわれます。

1Fステートメントの多重化 (ネスティング)

IF~THENを多順(ネスト)にすることができます。その限度は行の長

さによって制限されます。

IF A=B THEN IF B=C THEN PRINT"A=C"

IF~THEN〈ライン・ナンバー〉を実行した場合、指定されたライン・ナンバーの文がなければ、\*UNDEF'D STAEMENT"エラーが発生します。

20 IF LITHEN GET LIPRINI L

30 GOTO10

この場合、ライン・ナンバー2 $\emptyset$ では、もし $I \neq \emptyset$ ならば、キーボードから 1文字読みこもうとします(数字以外をキーボードから入力するとSY NTAX ERRORになります)。

# 2: 10 IF (I>10) AND (I<20) THEN DB=1980: GOTO300 20 PRINT"OUT OF RANGE"

300 END

この例では、もしIが10より大きくかつ20より小さい場合、DBの値が 計算され、ライン・ナンバー300へ分岐します。それ以外は、ライン・ナ ンバー20から以降へと実行を続けます。

ゴ : 整数以外の数はその内部表現(2進法)が必ずしも正確でないため(たとえば、.2は正確に表わすことはできません)、整数でない数を調べる場合はある範囲をとって調べることを、おすすめします。たとえば、変数Aの計算結果が1.0になった場合、1かどうかを調べる確かな方法は

IF ABS (A-1.0) <=1.0E-6 THEN...

です。もしこの結果が真なら、13<sup>6</sup>分の1の精度でAは1に等しいことになります。

#### 2.14 INPUT

議日日:

**書式:** INPUT ["〈プロンプト〉"; ]〈変数名〉

目的: プログラム実行時にキーボードから入力することを可能にする。

INPUTステートメントが実行されると、プログラムの実行は停止し、データの入力を待っている疑問符(?)がプリントされ、カーソルが点滅します。もし"〈プロンプト〉"がある場合は、疑問符の前にそれがプリントされます。そこで、キーボードからデータを入力して下さい。入力されたデータは、〈変数名〉で与えられた変数に代入されます。入力データの数はリスト中の変数名の数と一致していなくてはなりません。各データの項目はコンマで区切ります。もしINPUTステートメントで指定した個数よりも少ない数のデータしか入力されなかった場合には、さらに2つの疑問符を表示してもっとデータの必要なことを示します。また多すぎる個数のデータが入力された場合、またデータとしてコンマ(,),コロン(:)を入力した時、「?EXTRA IGNORED」というメッセージをプリントし、余分なデータ、コンマおよびコロンは無視して、プログラムの実行を継続します。

〈変数名〉は、(添字付きを含み)数値変数名でも文字数名でもかまいませんが、入力される各データは、変数名で指定した型に一致しなくてはなりません。INPUTステートメントに対する文字 (ストリング)の入力は、クォーテーション (″) で囲む必要はありません。

一度に入力できるデータの長さは、最大88文字までです。しかし、プロンプト (?+1スペース) の分が最低2文字必要なので実際は86文字です (リターンキーを押すと、それも1文字分にかぞえます)。

次にもし入力されたデータの型が、対応する変数名のものと一致しない場合は、\*?REDO FROM START"のメッセージがプリントされ、もう一度繰り返しデータの入力を求め、まちがって入力した値は無視されます。

また、キーボードからリターンキーだけを押した場合、〈変数名〉の値は、INPUTを実行する前に、代入された値がそのまま残っています。

例 1: 10 INPUT X

20 PRINT X"ジジョウ ハ"X 1 2

30 END

READY.

RUN

? 5 (画面に?があらわれ、カーソルが点滅したらキーボードよ り5を入力して下さい)

5 ジジョウ ハ 25

14 2 : 10 PI=3.14159265

29 INPUT"ハンケイ": R

30 A=PI \*R 1 2

40 PRINT"メンセキ": A

50 PRINT

69 GOTO29

RUN

ハンケイ?7.4 (キーボードより7.4を入力して下さい)

メンセキ 172.033614

ハンケイ?

į

19 3 : 10 INPUT A, B\$, C

2 15 INPUT#

曹式: INPUT#〈ロジカルファイル番号〉、〈変数名〉

目的: ファイルからデータを読みこみ、プログラムの変数に代入する。

説明: 〈ロジカルファイル番号〉は、データ入力のために、OPEN (2.21参照) によりオープンされたファイルに使われた番号を用います。〈変数名〉は、ファイルのデータを割りあてる変数名です。人力するデータの型は、〈変数名〉により指定された型と一致しなくてはいけません。INPUT#ステートメントでは、INPUTのように疑問符がプリントされることはありません。

ファイル中のデータは、INPUT文に対して入力するデータと同じ型になっていなくてはなりません。数値の場合は、先に続く空白(スペース)、キャリッジリターン、ラインフィードは無視されます。つまり、空白、キャリッジリターン、ラインフィード以外の最初の文字が、数値データの始まりとされるわけです。数値は空白、キャリッジリターン、ラインフィード、またはコンマによって区切られます。型が異なっていた場合は、\*?REDO FROM START のかわりに、\*?FILE DATA ERROR がプリントされます。EOIまたはタイムアウトによりINPUT#は1個のデータの入力を終わり、次へすすみます。

INPUT#で1回で入力できるデータの最大の長さは88文字です。

例 : 2.21 OPENの項を参照して下さい。

## 2.16 LET

**書式**: [LET](変数名)=(式)

目的: 〈式〉であらわされる値を変数に代入する。

説明: LETはなくてもかまいません。つまり、式の値を変数名に代入するに

は、等号だけでよいのです。

例: 10 LET A=23

20 LET B=2312

30 LET C=23†4

40 LET SM=A+B+C

は

10 A=23

20 B=231 2

30 C=231 4

40 SM = D+E+F

としても同じです。

#### 2.17 LIST

**書式 1**: LIST [〈ライン・ナンバー〉]

**書式 2**: LIST [〈ライン・ナンバー〉] - [〈ライン・ナンバー〉]

目的 : メモリー内にあるプログラムの全部、または一部のリストを出力機器

(たとえば、ディスプレイ、プリンターなど)に出力する。

### 説明 : 書式1)

1. 〈ライン・ナンバー〉が指定されない場合は、プログラムの全部の リストを出力します。

2. 〈ライン・ナンバー〉が指定されている場合、その番号に該当する 行のリストだけを出力します。

#### 書式 2)

1. 最初の〈ライン・ナンバー〉だけが指定された場合、その行から 最後までのリストが出力されます。

2.2番目の(ライン・ナンバー)だけが指定された場合、プログラ

ムの先頭から指定された行までのリストが出力されます。

3 (ライン・ナンバー)が両方共指定されている場合は、指定された 範囲のリストが出力されます。

LISTコマンドを実行し終わると、BASICは、つねにコマンドレベル に戻ります。

LIST : 書式1) [41]

プログラム全部のリスト

LIST500

ライン・ナンバー5000のリスト

書式2) LIST50ライン・ナンバー50から最後までのリスト

LIST-100

- 最初からライン・ナンバー1個までのリスト LIST59-100 ライン・ナンバー50から100までのリスト

2.18 LOAD

書式: LOAD("ファイル名") [. (デバイス番号)]

目的 外部記憶装置からメモリーにプログラムを読みこむ。

:20日 : (ファイル名)は、プログラムが、SAVE (2.28参照)された時に、 使用された名前です。

> デバイス番号を指定しない場合は、自動的に1(=カセット)が割りあて られます。

> LOADコマンドは〈ファイル名〉で指定したプログラムを読みこむまえ にメモリー内に現在あるすべての変数、プログラムを消去し、かつオー プンされていたファイルをすべてクローズします\_

> もしLOADコマンドをプログラム中で実行した場合はオープンされて いたファイルはそのままオープンされたままで、新しいプログラムを読 みこみ (=LOAD)、実行開始 (=RUN) します。

このようにしてLOADコマンドはいくつかのプログラム(または同じ プログラムのいくつかのセグメント)を連結して実行するのに使えます。 また、この場合、各プログラム間(ないしはセグメント間)で変数は消 去されませんので、データの受け渡しをおこなうことができます。

(ファイル名) および(デバイス番号) の指定がない場合は、カセット から1番最初にみつけたプログラムファイルを読みごみます。

例: LOAD "BASIC BASIC"

この場合、デバイス番号=1 (カセット) の機器から\*BASIC BASIC"という名前のプログラムファイルをメモリーに読みこみます。

2.19 NEW 書式: NEW

**目的**: 現在メモリー内にあるプログラム、およびすべての変数を消去する。

説明: 新しいプログラムを入力する前に、メモリーをクリアする時に、コマンドレベルでNEWを入力します。
BASICは、つねにNEWを実行した後、コマンドレベルに戻ります。

2.20 ON~GOSUB, ON~GOTO

 $ON\langle 式 \rangle GOTO\langle ライン・ナンバー \rangle [, \langle ライン・ナンバー \rangle...]$ 

目的: 〈式〉の評価により得られた値に基づき、指定されたいくつかのライン・ ナンバーの該当する行の1つに分岐する。

説明: 〈式〉の値が、どの行に分岐するかを決定します。たとえば〈式〉の値が 3であったなら、3番目に書かれた行が、分岐先となります(もし値が 整数にならない時は、小数部分は切り捨てられます)。

ON~GOSUBステートメントでは、各ライン・ナンバーは、サブルーチンの最初の行のライン・ナンバーでなければなりません。

もし〈式〉の値が負になった場合、\*ILLEGAL QUANTITY"エラーが発生します。またもし、値が、Øまたは、ライン・ナンバーの個数より大きい場合には、次の行へプログラムの制御が移り、エラーは起こりません。

例: 10 ON A-1 GOTO 20, 30, 40, 50

この場合、

A = 4 ならば、プログラムの制御は、ライン・ナンバー4 $\emptyset$ に移ります。 A = 6 ならば、制御は、ライン・ナンバー1 $\emptyset$ の次の行に移ります。

#### 2.71 OPEN

★式: OPEN〈ロジカルファイル番号〉〔,〈デバイス番号〉〔,〈セカンダリィーアドレス〉〔, "〈ファイル名〉"〕〕〕

員的: I/Oチャネルとシリアルバス、または内部デバイス間の連絡を確立する。

IN明: ロジカルファイル番号は次の範囲内で指定しなくてはいけません。 1 ≦ロジカルファイル番号≦255

デバイス番号を指定しない場合は、自動的に1(=カセット)が割りあてられます。セカンダリィアドレスおよびファイル名を指定しない場合は、ないものとみなされます。シリアルバスを使用した場合は、GET#、INPUT#、およびPRINT#が送られるのといっしょに必ずセカンダリィアドレスと、デバイス番号も送られます。

ファイルのタイプは、もし $^{\circ}$ S''(シーケンシャル)が指定されなければ、プログラムファイルになります。また、シーケンシャルファイルの場合( $^{\circ}$ S''で指定) $^{\circ}$ W''(ライト)が指定されなければ、読みこみ用のREAD(リード)ファイルとして扱われます。

ファイルは、カセット (デバイス番号=1)、プリンター (デバイス番号=4)、ディスプレイ (デバイス番号=3)、およびキーボード (デバイス番号=0) 対してオープンすることができます。

カセットテープを使用した場合

セカンダリィアドレスは

- カセットテープからファイルを入力します(読みこみ)。
- 1 カセットテープにファイルを出力します(書きこみ)。
- カセットテープにファイルを出力します(書きこみ。ただし、ファイルの最後にエンド 'オブ" テープ=EOTを書きこみます)。

#### です

例: 10 OPEN2.1.1. "DATAFILE"

- 20 FOR I= 1 TO10
- 30 PRINT # 2. CHR\$ (1±48)
- 40 NEXT
- 56 CLOSE 2
- 66 PRINT "REWIND TAPE"
- 70 PRINT "PRESS ANY KEY TO READ"
- 80 GET OS IF OS = "THEN 80
- 90 OPEN2, 1, 0"DATAFILE"

199 FOR I= 1 TO10

110 INPUT#2, A\$

120 PRINT VAL (AS)

130 NEXT

140 CLOSE 2

#### 2.22 POKE

**書式**: POKE I,J(Ø≦I≦65535,Ø≦J≦255)

ただし」および」は整数で表記

**目的**: メモリーの指定した番地(=I)に1バイトのデータ(=J)を書きこむ。

説明: 整数 I は書きこむメモリーの番地です。J は書きこむデータで Ø から255

の範囲でなくてはなりません。また【はりから65535の範囲です。

POKEと逆の働きをする関数にPEEKがあります。PEEKの引数は、1 バイトのデータを読みだしたいメモリーのアドレスです (3.13参照)。

POKEとPEEKは、効率的なデータの格納、機械語サブルーチンの書き こみや、機械語サブルーチンとのデータや計算結果などの受け渡しなど に役立ちます。

例: 10 POKE 4096.1

この場合4096番地に1が書きこまれます。

2.23 PRINT, PRINT#

**書式:** PRINT [#〈ロジカルファイル番号〉,][〈式〉....]

目的: データをディスプレイ、または指定したチャネルに出力する。

説明: 〈式〉が省略された場合、1行ブランク (空臼) が出力されます。〈式〉が

ある場合、数値や文字 (ストリング) が出力されます。文字の場合はク

ォーテーション(")で囲む必要があります。

プリント位置

句読点で各項目を区切ることにより、項目を区切って出力することができます。つまりBASICでは、88文字 (両面での4行分になります)を8個のゾーンに区切り、各項目をコンマ(,)で区切った場合、次のゾーン

の先頭から次の値が出力されます。セミコロン(;)で区切った場合は、前の値に続いて次の値が出力されます。項目間にスペースを1個以上いれても同様の結果が得られます。もし、〈式〉の最後が、コンマやセミコロンで終った場合、次のPRINTステートメントは、ひきつづき同じ行に出力されます。

コンマやセミコロンがない場合は、行の最後に自動的にキャリッジリタンがつけられ改行します。もし両面への出力が22文字をこえた場合、自動的に次の行へ出力されます。

数値がプリントされると、そのうしろにスペースが1個挿人されます。また数値の場合、正の数は、数値の前にスペースが1個、負の場合はマイナス符号がつけられます。数値は有効桁数の範囲内であれば、そのままで、そうでなければ、指数形式で表示されます。ある数値Xが $\emptyset$ X<、 $\emptyset$ 1 であるならば、X の型には関係なく、指数形式で表示されます。

キャリッジリターン (=( $\emptyset$ D)  $_{16}$ )は、各PRINT#ステートメントの最後に自動的につけられます。しかし、〈式〉の最後にセミコロン(;) をつけることによって避けることができます。

20 PRINT X+7, X-7, X(-7), X $\uparrow$ 7

30 END

RUN

10 -4 -21 2187

この場合、出力する各項目をコンマで区切っているため、各項目は、それぞれのゾーンの先頭から出力されます。

例 2: 10 INPUT X

20 PRINT X "ジジョウ="X † 2

3Ø PRINT X "サンジョウ="X † 3

40 PRINT

50 GOTO10

RUN

? 9 (9を入力します)

9 ジジョウ= 81.0000001

9 サンジョウ= 729.000001

? 8 (8を入力します)

8 ジジョウ= 64

8 サンジョウ= 512

この場合、ライン・ナンバー2gの最後はセミコロンをつけることにより、次のPRINTステートメントが、同じ行に出力されます。ライン・ナンバー4gでは、1行、ブランクを出力します。

(注) 9の2乗、3乗を計算した結果が、机上での計算結果と異なっています。これはVICがもっているアルゴリズムの結果の誤差です。たとえば、9の2乗(9↑2)を計算する場合、次のようになります。

2 loge 9 =xとすると

9 † 2 はexでもとめられます。

∴e×=81.00000001

(VICではe=2.71828183, 有効桁数は9桁のため以上の結果がえられ)ます。

例3: 10 FOR X=1TO5

20 J=J+2

30 K=K+4

4Ø ? J:K:

50 NEXTX

RUN

2 4 4 8 6 12 8 16 10 20

この場合、セミコロンにより、数値は、前の値にひき続いて(ただし、 符号分としての1スペースおよびおわりの1スペースが必ず出力されま すが)出力されます。ライン・ナンバー40では?がPRINTのかわりに使わ れています。ただし、LISTで、プログラムのリストをとりなおしますと、 今度はPRINTと表示されます。

2.24 READ

書式: READ (変数名)[,(変数名)...]

目的: DATAステートメントよりデータを読み、変数に割りあてる(2.5参照)。

説明: READステートメントは、必ずDATAステートメントと組み合わせて伸

わなければいけません。READステートメントはDATAステートメントのデータを1対1対応の方法で変数に割りあててゆきます。READステートメントの変数は、数値変数でも、文字変数でもかまいません。しかし読み出された値と変数の型は一致していなければなりません。もし一致していない場合は、?SYNTAX″エラーが発生します。

1つのREADステートメントが、1つ以上のDATA文を参照したり(複数の場合は順番に参照されます)、またいくつかのREADステートメントが1つのDATAステートメントを参照することができます。もし、〈変数名〉の個数がDATAステートメントの個数を越えてしまった場合は、\*OUT OF DATA″エラーが発生します。

指定された変数の数が、DATAステートメントのデータの個数よりも少ない場合には、続いてある次のREADステートメントが続きのデータから読み始めます。もしREADステートメントが、それ以上なければ、残りのデータは無視されます。

DATAステートメントの先頭からもう一度読みなおす場合はRESTORE (2.26参照)を使用して下さい。

例1: 1g FOR I=1TO1g

26 READ A (I)

30 NEXT

40 DATA7.06,4.19,3,3.98,1

5Ø DATA2,3,5.Ø9,7,1,8,4

÷

この場合、DATAステートメントからデータを読み、配列Aに代入します。実行するとA(1)は7.%6, A(2)は4.19と、各要素に順番に割りあてられます。

例 2: 10 PRINT "CITY ZIP"

20 READCS, Z

30 DATA "SANTA CLARA", 95050

40 PRINTCS:Z

RUN

CITY ZIP SANTA CLARA 95050

この場合、文字と数値データを読みこんでいます。

#### 2.25 REM

書式: REM (コメント)

目的: プログラム中に注釈、コメントを入れる。

説明: REMステートメントは実行はされませんが、プログラムのリストを取る と入力した内容 (カタカナのコメントは"で囲みます)がそのままリスト されます。REMステートメントの行へGOTOやGOSUBにより分岐する ことができます。この場合REMステートメントの後にくる最初の実行

文から実行を開始します。

例: 1000 REM CBM BASIC MANUAL

110 REM "コモドール ジャパン"

2.26 RESTORE 書式: RESTORE

目的: DATAステートメントを最初から読めるようにする。

例: 10 READ A.B.C

20 RESTORE

30 READ D, E, F

40 DATA57, 80, 91

-

この場合実行すると、変数AとDには57が代入されます。

#### 2.27 RUN

書式: RUN[〈ライン・ナンバー〉]

目的: 現在メモリー内にあるプログラムの実行を開始する。

説明: 〈ライン・ナンバー〉を指定すると、その行から実行がはじまります。指

定のない場合には、最も小さいライン・ナンバーから実行がはじまりま

す。

次の場合に、プログラムは実行を終了し、BASICはいつでもコマンドモードに戻ります。

ELY実行するライン・ナンバーが、もう、ない場合

(2) ENDまたはSTOPステートメントが実行された場合

③プログラム実行中に、何かエラーが発生した場合

28 SAVE

| 曹武: SAVE["〈ファイル名〉"[,〈デバイス番号〉[,〈コマンド〉]]]

目的: BASICプログラムファイルに書きこまれる(セーブする)。

説明: デバイス番号を指定しない場合は、自動的に1(=カセット)が割りあて られます。

コマンドが�の場合……プログラムのセーブ終了後テープの最後にエン

ドオブテープ(EOT)が書かれません。

コマンドが登以外の場合……セーブ終了後、エンドオプテープ (EOT)

が書かれます。

 $\langle ファイル名 \rangle$ を指定しない場合は、自動的にヌルコード(=( $\emptyset\emptyset$ ) $_{16}$ ) が割りあてられます。

例: SAVE

SAVE "COMMODORE"

SAVE (A\$)

2 29 STOP

書式: STOP

目的: プログラムの実行を停止してコマンドレベルに戻る。

説明: STOPステートメントはプログラムの実行を停止するためにプログラム のどこで使用してもかまいません。

のここで使用してもかまいません。

STOPを実行すると次のようなメッセージが出力され実行を停止します。

BREAK IN nnnn ライン・ナンバー

ENDステートメントとちがい、STOPステートメントは、ファイル をクローズしません。

STOPが実行されるとBASICは必ずコマンドレベルに戻ります。またCONTコマンド (2.4参照) により実行は再開されます。

例: 10 INPUT A, B, C

 $20 \text{ K} = (A+3)/2: L=B \times 3$ 

30 STOP

40 M= C X K+100 : PRINT M

RUN

? 1,2,3 (キーボードから入力します)

BREAK IN 30

READY.

PRINT L (ダイレクトモードで入力します)

6

READY.

CONT (ダイレクトモードで入力します)

196

READY.

### 2.30 SYS

書式: SYS (変数名) [((引数),...)]

目的: 機械語サブルーチンを呼びだします(3.27 USR関数も参照して下さい)。

説明: 〈変数名〉は、機械語ルーチンのメモリー上でのスタート番地です。〈変数

名〉として配列変数名は使用できません。

〈引数〉は、機械語のサブルーチンへ受け渡す引数です。

例: 199 JA=39299

110 SYSJA

2.31 VERIFY

**書式**: VERIFY["〈ファイル名〉"[,〈デバイス番号〉]}

目的: メモリー内のプログラムとテープ上のプログラムファイルとの内容を比

較してその違いを教える。

説明: デバイス番号が指定されない場合は、自動的に $1 (= \pi \pi + \pi)$ が割りあてられます。 $\langle 7\pi \pi \pi \pi \rangle$  を指定しない場合は、 $\pi \pi \pi \pi \pi \pi \pi \pi$ 

てられます。(ファイル名)を指定しない場合は、ヌルコード (=(ØØ)<sub>16</sub>)が割りあてられます。カセットを使用する時のみファイル名がなくても

かまいません。

例: VERIFY "KFILE"

PRESS PLAY ON TAPE # 1

OK

VERIFYING

VERIFY ERROR

READY.

2 32 WAIT

目的: コンピューターの入力ポートをモニター(監視)する間、プログラムの実行を停止する。

説明: WAITステートメントを実行すると、指定した番地 (アドレス)のビットパターンが指定した状態になるまでプログラムの実行が中断されます。 指定した番地から読みこんだデータと整数表現JとのEOR (排他的論理和) の結果とIのAND (論理積)が取られます。もしその結果が Øなら、BASICは、もう一度指定した番地の状態を読み込み同じ操作を繰り返します。もし結果が Øでなければ、プログラムの実行は次の文に移ります。 Jが省略された場合は、 Øとみなします。

[₩] : 10 WAIT37151,64,64

この場合、カセットのキーを押すまで、プログラムの実行を中断しています。

E: WAITステートメントの実行により、無限ループにはいってしまう場合 があります。その場合には、コンピューターをリセット (例、電源OFF) しなければなりません。

# 第3章 CBM BASICの関数

この章では、CBM BASICの持っている組み込み関数について、以下の形式で 説明します。これらの関数は、どのプログラムでも、何の定義の必要もなく使用 することができます。

**書式**:関数の正しい書式 (フォーマット) を示します。関数に渡す引数 (アーギュメント) については、下記を参照して下さい。関数にも省略形が使えます (付録Bを参照して下さい)。

機能:その関数の機能を示します。

例 :その関数の使用法を示すプログラム例をあげます。

関数に渡す引数 (アーギュメント) は、つねにカッコ ( ) で囲みます。 この章での関数の書式で、引数は次のように表現します。

X 、 Y 任意の数式をあらわします。

I , J 任意の整数をあらわします。

X\$ , Y\$ 任意の文字列をあらわします。

整数が必要な場合に、浮動小数点形式であらわされた数値が与えられた場合、 BASICは、小数部分を切り捨て、整数部分のみを使います。

## CBM BASIC関数の一覧表

| 関数      |      |         | 結果    |
|---------|------|---------|-------|
|         |      | ニューメリック | ストリング |
| ABS     | 3. 1 | X       |       |
| ASC     | 3. 2 | X       |       |
| ATN     | 3. 3 | X       |       |
| CHR\$   | 3. 4 |         | X     |
| cos     | 3. 5 | X       |       |
| EXP     | 3.6  | X       |       |
| FRE     | 3. 7 | Х       |       |
| INT     | 3.8  | X       |       |
| LEFT\$  | 3. 9 |         | X     |
| LEN     | 3.10 | Х       |       |
| LOG     | 3.11 | X       |       |
| MID\$   | 3.12 |         | X     |
| PEEK    | 3.13 | X       |       |
| POS     | 3.14 | Х       |       |
| RIGHT\$ | 3.15 |         | X     |
| RND     | 3.16 | X       |       |
| SGN     | 3.17 | X       |       |
| SIN     | 3.18 | X       |       |
| SPC     | 3.19 |         | Х     |
| SQR     | 3.20 | Х       |       |
| STATUS  | 3.21 | X       |       |
| STR\$   | 3.22 |         | X     |
| TAB     | 3.23 |         | X     |
| TAN     | 3.24 | X       |       |
| TIME    | 3.25 | X       |       |
| TIME\$  | 3.26 |         | X     |
| USR     | 3.27 | X       |       |
| VAL     | 3.28 | X       |       |
|         |      |         |       |

### 3.1 ABS

書式; ABS(X)

機能: 式 X の絶対値を与えます。

例: PRINT ABS (7\*(-5))

35

### 3.2 ASC

書式: ASC(XS)

機能: ストリングXSの最初の文字のASCII(アスキー)コードを与えます

(ASCIIコードについては付録Dを参照)。

例: 10 X\$="V!C"

20 PRINT ASC(X\$)

RUN 86

ASCIIコードから文字への変換については、CHR\$関数 (3.4) を参照して下さい。

### 3.3 ATN

書式: ATN(X)

機能: Xのアークタンジェントの値(ラジアン単位)を与えます。演算結果は、 ー元/2から元/2の範囲になります。式Xはどのような型の数値であっても

かまいませんが、ATNの演算は常に浮動小数点形式でおこなわれます。

例: 10 INPUT X

20 PRINT ATN(X)

RUN

? 3 (キーボードより3を入力します。)

1.24904577

1.4 CHR\$

畫式: CHR\$(I)

機能: ASC(I(アスキー) コードが I である文字を与えます (ASCII コードにつ

いては付録Dを参照)。通常、CHR\$は特別な文字を出力するのに用いられます。たとえばCHR\$ (147)を出力することにより、画面をクリア

(消去)し、カーソルをホームポジションに戻すことができます。

例: PRINT CHR\$(65)

Α

文字をASCIIコードに変換するのは、ASC関数(3.2)を参照して下さい。

3.5 COS

書式: COS(X)

機能: X (ラジアン単位) のコサインの値を与えます。COS (X) の演算は

浮動小数点形式でおこなわれます。

|M| : 10 X=2 \*COS(1)

20 PRINT X

RUN

1 08060461

1.6 EXP

商式: EXP(X)

機能: eをX乗した値を与えます。Xは88.∅2969191より小さいか等しい値でな

ければいけません。

もしEXP関数がオーバーフローした場合、\*OVERFLOW"エラーが発

生します。

14 : 10 X=3

20 PRINT EXP(X-1)

RUN

7.3890561

### 3.7 FRE

書式: FRE(X)

機能! メモリー内でBASICが未使用のバイト数を与えます。FRE関数で使う

引数Xはダミーですので、任意の数値を使用して下さい。

例: PRINT FRE(0)

3553 (その時に未使川のバイト数が表示されます)

### 3.8 INT

書式: INT(X)

機能: Xより小さいか等しい最大の整数を与えます。

例: PRINT INT(89.72)

89

PRINT INT(-23.17)

-24

### 3.9 LEFT\$

書式: LEFT\$ (XS.1)

機能: X\$の左側からI個の文字列を与えます。IはØから255までの範囲の数

でなければなりません。

もしI がX \$ の桁数(=L E N (X \$)) より大きければ、X \$全体を結

果として与えます。

もし「がりならば、ヌルコード ((例)16, 長さは例) を与えます。

10 AS="COMMODORE JAPAN"

20 B\$=LEFT\$(A\$.9)

36 PRINTBS

RUN

COMMODORE

MID\$、RIGHT\$関数も参照して下さい。

3.10 LEN

書式: LEN(X\$)

機能: X\$を構成する文字の長さ(桁数)を与えます。プリントされない文字

や空白 (スペース) も数えます。

₩ : 10 X\$="COMMODORE JAPAN"

20 PRINT LEN(XS)

RUN 15

J.11 LOG

書式: LOG(X)

機能: Xの自然対数を与えます。Xは必ずøより大きくなければなりません。

| PRINT L.OG (37/5)

2.00148

J 12 MID\$

#式: MID\$(XS,I[,J])

機能: X\$の左側よりI番目からJ個の文字列を与えます。 IおよびJは、Øから 255までの範囲の数でなければいけません。もしJが省略されていたり、

I番目の文字より右にJ個より少ない文字しかない場合は、I番目より右側の文字全部を関数の値として与えます。

\* I I I EN/Ve) ~ + h v e n k w x i L

もし!>LEN(X\$)、つまり X \$ の桁数が! より小さい場合は、関数の値として、ヌルコード(=(ØØ)16, 長さØ)を与えます。

例: 10 A\$="GOOD"

20 B\$="BY LUCK"

36 PRINT A\$:MID\$(B\$,3,5)

RUN

GOOD LUCK

LEFT\$、RIGHT\$も参照して下さい。

### 3.13 PEEK

書式: PEEK(I)

機能: I番地のメモリーの内容を読みだした値(例から255までの10進整数)を

与えます。 I は Ø から65535までの範囲の数でなければなりません。PE

EKは、POKE (2.22参照) の逆の働きをする関数です。

例: A=PEEK (4096)

3.14 POS

書式: POS(X)

機能: 画面上のカーソルの現在の横位値を与えます(左端が Ø)。引数 X は ダミ

ーです。

例: 166 PRINT" ";POS(X)

RUN 7

3.15 RIGHT\$

書式: RIGHT\$(X\$,I)

機能: X \$ の右側から「個の文字列を与えます。

もし、I = > LEN(X\$)、つまりX\$の桁数とIが等しいまたは大

きいならば、結果はX\$全部を与えます。

もし、 $I=\emptyset$ ならば、ヌルコード (=( $\emptyset\emptyset$ )<sub>16</sub>, 長さは $\emptyset$ ) を与えます。

例: 10 AS="COMMODORE JAPAN"

20 B\$=RIGHT\$ (A\$,5)

30 PRINTB\$

RUN JAPAN

MID\$, LEFT\$も参照して下さい。

### 3.16 RND

書式: RND(X)

機能: Xの正負 Øにより1、-1、 Øを与えます。

X < Ø の場合、その数字に割りあてられた1個の乱数を常に与えます。

X=Øの場合、使用のたびに同一の順序の乱数系列を与えます。

X>Øの場合、使用のたびに新しい乱数系列を与えます。

例: 10 FOR I= 1 TO 3

20 PRINT INT (RND(1) \*100);

30 NEXT

RUN

66 1 90

(ここで表示される数は乱数ですので必ずしも左の) 数と同じではありません。

### 3.17 SGN

書式: SGN(X)

機能: Xの正負 $\emptyset$ により1、-1、 $\emptyset$ を与えます。

X > Ø の場合。SGN (X) = 1

 $X = \emptyset$ の場合、SGN  $(X) = \emptyset$ 

 $X < \emptyset$  の場合、SGN (X) = -1

#### 图: 5 INPUTX

10 ON SGN(X)+2 GOTO100.200.300

100 PRINT "X<0":END

200 PRINT "X=0":END

300 PRINT "X>0":END

この場合、X<Øの時、ライン・ナンバー100へ、

 $X > \emptyset$ の時、ライン・ナンバー30 $\emptyset$ へ、

 $X = \emptyset$ の時、ライン・ナンバー2000へ

分岐します。

### 3.18 SIN

書式: SIN(X)

機能: X (ラジアン単位) のサインの値を与えます。SIN(X)の演算は、浮動

小数点形式でおこないます。

例: PRINT SIN(1.5)

.997494987

3.19 SPC

書式: SPC(I)

機能: I個の空白(スペース)で構成される文字列を与えます。SPC関数は、

PRINTと共にのみ使用されます。 I は Ø から255までの範囲の数でなけ

ればいけません。

例: PRINT"GOOD" SPC(10) "LUCK"

GOOD LUCK

3.20 SQR

**書式**: SQR(X)

機能: Xの平方根を与えます。ただしXは必ず正またはØでなければなりませ

 $h_{\circ}$ 

例: 16 FOR X=2TO16STEP2

29 PRINT X; SQR (X)

30 NEXT

RUN

2 1.41421356

4 2

6 2,44948974

8 2.82842713

10 3.16227766

#### 1 21 STATUS

書式: ST

機能: カセット、画面(スクリーン)、キーボード、シリアルバスなどで最後に 人出力操作をおこなった時のコンピューターの状態を与えます。

| マテー <b>タス</b><br>レット位置 | ステータス値 | カセットリード   | シリアルバスリード/<br>ライト | カセット<br>ベリファイ・ロード |
|------------------------|--------|-----------|-------------------|-------------------|
| Ø                      | 1      |           | リスナータイムアウト        |                   |
| 1                      | 2      |           | トーカ・タイムアウト        |                   |
| 2                      | 4      | ショートブロック  |                   | ショートブロック          |
| 3                      | 8      | ロングブロック回復 |                   | ロングブロック           |
| 4                      | 16     | 不能なリードエラー |                   | ミスマッチ             |
| 5                      | 32     | チェックサムエラー |                   | チェックサムエラー         |
| 6                      | 64     | エンドオブファイル |                   |                   |
| 7                      | -128   | エンドオブテープ  | 当番機器なし            | エンドオブテープ          |

### $\sqrt{3} - \sqrt{5} = 4$

テープからプロックを読み込んでいる時、読み込まれるべきバイト数に達しないうちにスペース音が入ってきたことを示します。原因としては、LOADされるはずの短いプログラムファイルがデータファイルとして読まれた場合が考えられます。

### ロングブロック) ST=8

テープからのブロック読み込み時に、読まれるべきバイト数を超えたにもかか わらず、スペース音が来ない場合、原因としては、長いプログラムファイルをデータファイルと誤まって読んだ場合が考えられます。

### 回復不能なリードエラー) ST=16

原因として冗長ブロックのうちの最初のブロックに31個以上のエラーがあった 場合、もしくは冗長ブロックと基本ブロックの双方同一ケ所にエラーがおこって 修復不可能な場合が考えられます。

### ·チェックサムエラー) ST=32

LOADや、データの読み込みの終了後、RAM中の各バイトについてチェックサムが計算され、入力デバイスから受け取ったバイトと比較されます。もし一致しなければ、チェックサムエラーとなります。

### (エンドオブファイル) ST=64

データテープに、エンドオブデータファイルマークがあるとSTに64がセットされます。

### (エンドオブテープ) S T = -128

**EOT**(エンドオブテープ)を読んだ場合、STに128がセットされます。

例: 19 OPEN 2

20 INPUT# 2, A\$

30 IF ST=0 OR ST=64 THEN50 (ST=0は動作が正常に

49 GOTO29

終了したことを示します)

50 PRINT A\$
60 CLOSE 2

#### 3.22 STR\$

書式: STR\$ (X)

機能: Xの数値を表わす文字列を与えます。

例 : 5 REM ARITHMETIC FOR KIDS

10 INPUT"TYPE A NUMBER": N

20 ON LEN(STR\$(N)) GOSUB30,50,70,90

25 END

30 PRINT"LENGTH= 1": RETURN

50 PRINT"LENGTH= 2": RETURN

70 PRINT"LENGTH= 3": RETURN

90 PRINT"LENGTH= 4":RETURN

VAL関数(3.28)も参照して下さい。

#### 3.23 TAB

★式: TAB(I)

機能: 画面 (スクリーン) またはターミナルの1桁目の位置まで空白 (スペース) をプリントします。もし現在のプリント位置が1を越えていれば、

 ${
m TAB}$ は無効です。 ${
m TAB}$  (  $\emptyset$  ) が左端になります。右端は画面の桁数

から1ひいた数が1になります。

「は夏から255までの範囲でなければなりません。また、TABは、P RINTステートメントでしか使用できません。 III : 10 PRINT"NAME"TAB (15) "HEIGHT": PRINT

20 READ A\$. B\$

3# PRINT A\$; TAB (15); B\$

40 DATA"BOB", "180.19"

RUN

NAME HEIGHT

BOB 180.19

3.24 TAN

書式: TAN(X)

機能: X (ラジアツ単位) のタンジェントの値を与えます。TAN(X)の演

算は、浮動小数点形式でおこなわれます。TAN関数がオーバーフロー

した場合は、\*OVERFLOW" エラーが発生します。

[M] : 10 Y=P \* TAN (X) / 3

1 25 TIME 會式: TI

機能: 内部タイマーを読むのに使用されます。約1/60秒 (1ジフィー) の単位

で内部タイマーの時間が与えられます。これは、実際の時計とはちがい

ます。

例: 10 A=TI

20 IF TI-A<120THEN20

30 PRINT" 2 SECONDS"

この場合TIは、約1/60秒毎に値がかわってくるために、プログラムを 実行しますとライン・ナンバー10でAにスタート時間をセットして、約 2秒後に、2SECONDS″をプリントします。 3.26 TIME\$ 書式: TI\$

内部タイマーを読んで、その結果を『HHMMSS"(HH:時, MM:分)。

SS: 秒) の文字列 (ストリング) で与えます。電源を投入した直後に内 部タイマーはすべて9999999にセットされます。したがって、内部タイマ ーに初期値を代入しない場合は、TISはコンピュータの使用時間をあら わします。

内部タイマーの値に初期値を与えるためには、INPUTステートメントと いっしょに、または、代入式(下記の例を参照)を用います。

例: 10 TIS="000000"

20 FOR I= 1 TO 1000 : NEXT

30 PRINT TIS

RUN

**9999991** (実行すると、約1秒後にこれが表示されます)

3.27 USR

- 書式: USR(X)

機能: メモリーの1番地、2番地にストアされている番地に、プログラムのコ

ントロールを移し、引数をフローティングアキュムレーターに格納します。

例 : 828番地にダイレクトモードでキーボードから、次のようにして機械語ル

ーチンを書いておきます。

POKE828, 169

POKE829, 01

POKE830, 141

. ONE 832,39 画面 (V-RAM) は77例2番地から開始しています。 POKE831,0 .

POKE833, 96

POKE1.60

POKE2.3

画面をクリアして、カーソルをホームポジションに戻して下さい。そし 7.

B=USR(@)

と入力します。

すると、画面の左上に \*A" の文字が出力されます。次に **PRINT B** と入力しますと **10** と表示されます。

1 28 VAL

当式: VAL (X \$)

機能: 文字列 (ストリング) X \$ の表わす数値を与えます。もしX \$ の最初の 文字が十、一、または数字でなければVAL(X\$) の値はØになります。

| 10 A\$="123" 20 B\$="+123" 30 C\$="-123" 35 D\$="ABC" 40 PRINT VAL (A\$) 50 PRINT VAL (C\$) 70 PRINT VAL (D\$) RUN 123 123 -123 0

> 数値を文字列 (ストリング) に変える場合は、STR\$関数 (3.22) を参 照して下さい。

# 第4章 CBM BASICのプログラム例

この章は、これまでに説明したCBM BASICのコマンド、ステートメントを用いて作成した、簡単なプログラム例を集めてみました。いずれも短かくて、かつ結果がスクリーンにすぐに表示されるもの、音が出力されるものばかりです。何はともあれ、プログラム・リストを見ながら、キーボードからプログラムを入力して下さい。それを実行させてエラーが発生した場合、エラーメッセージ・覧表などを参照して、原因を究明して下さい。また、これらのプログラムを参考にして、より楽しい、高度なプログラムをたくさん作って下さい。

### カーソル制御、反転指定、キャラクター色指定のプログラム

CBM BASICでは、カーソル制御、反転および反転解除指定、キャラクター色指定が、簡単にプログラムできます。コマンドは、ダブルクォーテーション (\*)で囲んで、用います。

|       |            | T                     |
|-------|------------|-----------------------|
| コマンド  | キー操作       | 画面にあらわれる文字            |
| 画面クリア | "CLR"      | (詞)                   |
| ホーム   | " HOME "   | ă                     |
| カーソル右 | " CRSR⇒ "  | NI .                  |
| カーソル左 | " CRSR⇔ "  | II.                   |
| カーソル下 | " CRSR ₽ " | 001                   |
| カーソル上 | " CRSR û " | 刀(图)                  |
| 一字追加  | "INST"     | (2)                   |
| 反 転   | "CTRL R"   | 2                     |
| 反転解除  | "CTRL Ø"   |                       |
| ブラック  | "CTRL 1"   | <b>■</b> ( <b>E</b> ) |
| ホワイト  | "CTAL 2"   | =                     |
| レッド   | "CTRL 3"   | H                     |
| シアン   | "CTRL 4"   | <b>▶</b> (目)          |
| マジェンタ | "[CTRL] 5" | <b>( )</b>            |
| グリーン  | "CTRL 6"   | 61                    |
| ブルー   | "[CTRL] 7" | <b>23</b>             |
| イエロー  | "[CTRL] 8" | ភា                    |

### 1. はじめまして

- 10 REM # HOW DO YOU DO ? # 20 30 40 REM A\$=" 對於表現事事事事 " 50 PRINT" " \* \* HOW DO YOU DO ? XX" 60 70 PRINT A\$" # " PRINT TAB(5)"[₫V ≝1" 80 PRINT TAB(5)" | | " 90 100 FOR I=1TO500: NEXT 110 PRINT A\$" " 120 PRINT TAB(5)"I ₩ I" 130 PRINT TAB(5)" | | " 140 FOR I=1TO500: NEXT 150 PRINT AS" # " 160 PRINT TAB(5)"[3V 集]" 170 PRINT TAB(5)" | | " 180 END
- VICぼうやがおじぎをします。
- )イン・ナンバー7 $\emptyset$ ~9 $\emptyset$ とライン・ナンバー15 $\emptyset$ ~17 $\emptyset$ は両方共、顔をあげた状  $\oplus$  です。
  - ライン・ナンバー110~130は頭をさげた状態です。
  - ライン・ナンバー100、140は、その間、画面を停止させます。

# 2. VIC SQUIGGLE (スクィグル)

```
CS=" FINE CT "
5
     PRINT".":
    DATA "|","-","|","|","|","|","|","
10
20
     DATA 1,0,5,6
   DATA Ø,1,4,3
30
40 DATA 3,6,2,0
50
    DATA 4,5,0,2
60
    DIM A$(5), B(5,5)
70 FOR I=0TO5
80 READ A$(I)
90
     NEXT
100
   FOR I=1TO4
110 FOR J=1TO4
120
   READ B(J.I)
130 NEXT
140
    NEXT
190
   T1=1
200 T2=1
210
   X == 20
226
   Y = 12
300
   REM * * *START
310 T1=4 \% RND (1)+1
320
   IF B(T1, T2)=Ø THEN31Ø
325
    GOSUB2000
330
    T2=T1
    ON T1 GOTO400,410,420,430
340
400 Y=Y-1:GOTO500
   Y=Y+1: GOTO500
410
420
   X=X+1:GOTO556
430
   X=X-1:GOTO556
500 IF Y<1 THEN Y=20: GOTO300
510
   IF Y>20 THEN Y=1:GOTO300
550
   IF X<1 THEN X=20: GOTO300
560
   IF X >20 THEN X=1:GOTO300
570
   GOTO300
2000 PRINT" 31":
2010 FOR I=1TOY
2020 PRINT" ":
2030 NEXT
2040 FOR I=1TOX
```

```
2050 PRINT" 1":

2055 NEXT

2060 PRINT" 1" "MID$(C$,RND(1) *7+1,1);

2080 PRINT A$(B(T1,T2)-1);

2090 RETURN
```

/イン・ナンバー2060を次のようになおすと、また、ちがった画面を楽しむこ ができます。

2060 PRINT MID\$ (C\$, RND(1) \*7+1,1);

# 3. レコーディング その1

1 REM \* \* \* \* \* \* \* \* \* \* \* \* 2 REM # RECORDING # 3 REMXXXXXXXXXXX 4 REM 10 INPUT"LENGTH": L 20 DIM S(L), L(L) 30 FOR I=1TOL 40 INPUT A.B 500 S(1) = A:L(1) = B60 NEXT 70 PRINT"WRITE OR RUN(W OR R) 7 " GET OS: IF OS="R"THEN110 90 IF O\$<>"W"THEN80 100 GOSUB200 119 POKE36878.15 120 FOR I=1TOL 130 POKE36874.S (I) 140 FOR J=@TOL(I) \* 300 150 NEXT : NEXT 160 FND 200 OPEN1.1.1 210 PRINT#1:L 220 FOR (=1TOL:PRINT#1.S()):PRINT#1.L() 230 NEXT: CLOSE1: GOTO70

音符のかわりにキーボードから、コードを入力して、VICに演奏させましょう。 人力したコードは、テープにセーブして残しておくことができます。

### 〈実行手順〉

- ① RUN RETURN
- ② 画面に LENGTH? と表示され、カーソルが点滅します。
- ③ キーボードから音符の数をいれます。 たとえば、次の場合



### 11と入れて下さい。

- 4) 次に、音のコードと長さをいれます。
  - ③の例の場合、

| $\emptyset, 2$ | RETURN |
|----------------|--------|
| Ø,1            | "      |
| 191,1          | **     |
| 191,1          | 4      |
| 198,1          | "      |
| 207,1          | 4      |
| 198,1          | "      |
| 207,1          | "      |
| 207,1          | 79     |
| Ø,2            | "      |
| Ø,4            | //     |

5) ③で指定した数だけコードと長さを入れると、画面は WRITE OR RUN (W OR R) ?

### と表示されます。

- R を押すと、④で作った音が流れてきます。
- W を押すと、PRESS PLAY & RECORD ON TAPEと表示されますので、テープをセットしてデータを書きこんで下さい。
- 6) 書きこみが終了すると⑤に戻ります。

参照:第2部付録Gサウンド

# 4. レコーディング その2

16 OPENI 20 INPUT#1.L 30 DIMS(L), L(L) 40 FORI=1TOL 50 INPUT #1, S(I), L(I)60 NEXT 70 PRINT"END OR RUN(E OR R) ?" 80 GETOS: IFOS= "R"THEN110 9# IFO\$ <> "E"THEN8# 100 END 110 POKE36878.15 120 FORI-ITOL 130 POKE36874,S(I) 140 FORJ=0TOL (1) \* 100 150 NEXT: NEXT

レコーディング その1で作成したテープを読んで音を出します。

① RUN RETURN

160 GOTO70

- ② PRESS PLAY ON TAPE と画面に表示されます。テープをセットして PLAY を押して下さい。
- ③ END OR RUN (E OR R)? と表示されます。
- ④ E を押すと、プログラムを終了します。R を押すと、音が出ます。終わった後、③へ戻ります。

# 5. VICオルガン

- 100 REM
- 110 REM VIC ORGAN
- 120 REM
- 130 DIM K\$(8), KY(9), MU(9)
- 140 FORI=1T09:READ KY(I):NEXT
- 145 FOR!=!TO9:READ\_MU(I):NEXT
- 150 DATA 0.56.1.57.2.58.3.59.4
- 155 DATA 63,70,76,79,85,89,93,95,0
- 160 CRS=" 3778 CBT "
- 180 FORI=1TO8
- 198 K\$(I)=MID\$(CR\$,I,I)+SP\$
- 2000 NEXTI
- 205 K\$(Ø)="3"+SP\$
- 210 VAS=" STATEMENTALE "
- 22# PRINT" 32 "
- 23Ø FORI=1TO8
- 240 PRINTVA\$:SPC((I-1) \*2) K\$ (0)
- 250 NEXT
- 260 A=PEEK (197): IFA=63THENGOSUB1000
- 270 FORI=1TO8
- 280 IF A=KY(1) THENJ=1:1=8:NEXT:GOTO300
- 290 NEXTI:GOTO260
- 300 POKE36878,15:PRINTVA\$;SPC((J-1) \*2);K\$(J)
- 305 POKE36874, (MU(J) + 128)
- 310 GOTO260
- 1999 POKE36874, 9: FORH=15TO9STEP1: POKE36878, H: NEXT: RETURN

### 1) RUN RETURN

- 2) 1を押すと、ドの音が出ます。
- 6を押すと、ラの音が出ます。

2 " V "

7 " > "

3 " \$ "

**Ø** / 音が消えます。

4 " 7r ' 5 " Y '

参照:第2部付録ドカラーコントロール、付録Gサウンド

# 6. 壁こわしゲーム

```
110 REM XVIC WALL DESTROY *
130 GOSUB620 REM TITLE WRITE
140 VR=7680
150 DEF FNA(X)=(V-1)X22+X+VR:REM VRAM ADDRESS
160 DEF FNB(X) = (PEEK (FNA(X+1)) = 32)
170 V=2:REM V=VERTICAL
180 SC$="0000":WA$="置顺":PD$="6":SP$=" 顺":
200 TIS="000000"
210 PRINT" DESCORE: 0000 TIME: 00": PRINTPDS
220 GOSUB410
230 GETAS
249 PRINT " S"; SPC (16): RIGHT$ (TI$, 2)
250 IFTI$ > "000100" THEN 500
260 IFA$=" "THEN230
270 IFA$="№"THENDT=1:GOSUB320:GOTO300
280 IFA$=""]"THENDT=-1:GOSUB320:GOTO300
290 IFAS=" "THENGOSUB360
300 IFWA=0THENGOSUB410
310 GOTO230
320 PRINT LEFT$(VA$,V):SP$:V=V+DT
330 IFV>20THENV=20
340 IFV<2THENV=2
350 PRINT LEFT$ (VA$, V):PD$:RETURN
360 POKE 36879,128+2
365 POKE FNA (Ø), 81: POKEFNA (Ø), 97
37Ø FORI= 1TO2Ø:POKEFNA(I),81:POKE FNA(I),32
380 IFFNB(I) =- 1THEN400
390 GOSUB460
400 NEXTI:POKE 36879,27:RETURN
410 Z$=" ":FOR!=1TO19
420 IFRND(1)>.7THENZ$=Z$+WA$:WA=WA+1:GOTO440
430 Z$=Z$+SP$
440 NEXTI
450 PRINT "国际"SPC (20) Z$:RETURN
46# POKEFNA (1),32:POKE FNA (I+1),42:POKE FNA (I+1),32
470 WA=WA-1:I=20:SC=SC+10
480 PRINT " ***** ": RIGHT$ (SC$+MID$ (STR$ (SC) .2) ,4)
490 RETURN
```

- 500 PRINT" ₩"SPC (16) " X X"
- 510 PRINT LEFT\$(VA\$.10)" 脚脚腳腳部 GAME OVER "
- 520 PRINT MAPRESS MRETURN TO START
- 530 GETA\$:IFA\$< >CHR\$(13)THEN530
- 540 RUN
- 55# DATA " MAIVIC WALL DESTROY."
- 560 DATA " ME KEYBORD COMMAND."
- 570 DATA """
- 586 DATA / 照照 CRSR UP E ..... MOVE UP.
- 590 DATA " MINI CRSR DOWN ... MOVE DOWN.
- 600 DATA "與翻譯各自印度實際...TO FIRE.
- 610 DATA " \* "
- 620 PRINT" TRIBUNE"
- 630 READ A\$:IFA\$="\text{\pi}"THENRESTORE:GOTO720
- 640 FORI-ITOLEN(A\$)
- 650 PRINT MID\$ (A\$,1,1)" \_\_\_\_";
- 66# FORJ=1TO5
- 670 GETB\$:IFB\$<>" "THENJ=5
- 680 NEXTJ
- 690 IFB\$<>" "THENI=LEN(A\$):NEXTI:GOTO760
- 700 NEXTI
- 710 PRINT" ": GOTO630
- 720 FORI=1TO500
- 730 GETA\$:IFA\$<>" "THENI=500
- 740 NEXTI
- 750 IFA\$=" "THEN620
- 760 RETURN

#### LET'S PLAY!

Li) ライン・ナンパー14 $\emptyset$ のVRは、スクリーンのスタートアドレスです。拡張メモリーの 8 Kまたは16 Kを使用している場合は、14 $\emptyset$  VR=4 $\emptyset$ 96に変更して下さい。

## 7. ひらがな

- 1 POKE51, Ø: POKE52, 19
- 2 POKE55, Ø:POKE56, 19
- 3 CLR
- 5 FORI=1TO40
- 10 READ A
- 20 POKE1023+4096+1,A
- 30 NEXT
- 40 GOTO100
- 50 DATA16,124,16,58,84,154,178,82
- 60 DATA0, 2, 129, 129, 129, 129, 65, 2
- 61 DATA24.0.24.36.4.4.4.8
- 62 DATA48, Ø, 12Ø, 16, 32, 84, 136, Ø
- 63 DATA32,248,34,33,120,164,164,68
- 100 POKE36867, PEEK (36867) AND254
- 105 POKE36866, PEEK (36866) OR128
- 110 POKE36869.253
- 140 END
- ① RUN RETURN と押して実行させます。
- ② @を押すと、画面に 'あ" が表示されます。
  - A \*\3"
  - B ″ <sup>\*</sup>う″

  - D " \*お" "

参照:第2部付録日ハイ・レゾリューション

# 8. 文字パターン作成プログラム

```
110 REM * MAKE PROGRAMMABLE CHARACTER X
120 REM *
130 REM * FOR VIC 1001
150 VR=7680
160 V$=" NT | W$="......
17# PRINT" DWHICH ?"
180 PRINT" IN (HIT NUMERIC KEY.)"
196 PRINT"與國新1篇..8 *8 DOT."
200 PRINT"國歌2豐..8 ※16 DOT."
210 GETA$:IFA$=""THEN210"
220 IFA$<>"1"ANDA$<>"2"THEN210
230 CH=0:IFA$="2"THENCH=1
240 CL=7:IFCH=1THENCL=15
250 PRINT"33":
260 J=8:IFCH=1THENJ=16
270 FORI=ITOJ:PRINTW$:NEXTI
280 X=VR:POKEX.(PEEK(X)OR128)
290 GETAS: IFAS=""THEN290"
300 IFAS=CHR$ (20) THEN590
310 IFAS=CHR$ (13) THEN650
320 FORI=1TO8
330 IFMID$(V$,I,1) = A$THENJ=II = 8 : NEXTI : GOTO360
340 NEXTL
350 GOTO290
360 ONJGOTO370,400,430,460,490,520,560
370 REM CURSOL DOWN
380 X0=22:Y%=Y%+1:IFY%>CLTHENY%=CL:X0=0
39# GOSUB63#:GOTO29#
400 REM CURSOL UP
410 X\emptyset = -22:Y\% = Y\% - 1:IFY\% < \emptyset THENY\% = \emptyset:X\emptyset = \emptyset
420 GOSUB630:GOTO290
430 REM CURSOL RIGHT
446 \times 6 = 1 \times \% = \times \% + 1 \times \% > 7 \text{ THENX } \% = 7 \times \% = \emptyset
450 GOSUB630:GOTO290
460 REM CURSOL LEFT
4761 \times 0 = -11 \times \% = \times \% - 11 \times \% = 0 \times \% = 0 \times \% = 0
489 GOSUB639:GOTO299
490 REM CURSOL HOME
```

```
500 \times \% = 0 \times \% = 0
```

- 510 POKEX, (PEEK (X) AND127) :X0=0:X=VR:GOSUB630:GOTO290
- 520 REM PATTERN CLR
- 530 X%=0:Y%=0:PRINT"图":
- 540 Y=8:IFCH=1THENY=16
- 550 FORI=ITOY:PRINTWS:NEXT:GOTO280
- 560 REM DOT!
- 570 POKEX.81:X0=1:X%=X%+1:IFX%>7THENX%=7:X0=0
- 580 GOSUB630 GOTO290
- 590 REM DELETE DOT
- 600 POKEX,46:X0=-1:X%=X%-1:IFX%<0THENX%=0:X0=0
- 610 GOSUB630 GOTO290
- 620 REM RVS SUBROUTINE
- 630 POKEX, (PEEK (X) AND127) :X=X+X0:POKEX, (PEEK (X) OR128)
- 640 RETURN
- 650 REM CALCULATE
- 660 PRINT" ≥ ":
- 670 Z=7:IFCH=1THENZ=15
- 680 POKEX, (PEEK (X) AND127)
- 690 FORI=VRTOVR+22 \*ZSTEP22
- 700 FORJ=0TOZ
- 710 IFPEEK (I+J) <>81THEN730
- 720  $Y = Y + 2 \uparrow (7 J)$
- 730 NEXTU
- 740 IFI=VRTHEN750
- 750 PRINTTAB (10) MID\$ (STR\$ (Y), 2):Y=0
- 76# NEXTI
- 776 PRINT :PRINT WE RUN TIT :END

7.のひらがなプログラムのライン・ナンバー50から63までのDATA文に書かれた数字を簡単に見つけることができます。

# 付 録

A:CBM BASICのコード表

B:CBM BASICの省略形

C:他のBASICからCBM BASICへの変換

D:エラーメッセージ一覧表

E:誘導関数

F:カラーコントロール

G:サウンド

H:ハイ・レゾリューション

# A: CBM BASICのコード表

BASICはコマンド、ステートメント、関数などを、1文字 (=1バイト)に変換してメモリー内部に格納します。変換するコードは次表の通りです。

| コード | BASICの表現 | コード | BASICの表現 | コード | BASICの表現  |
|-----|----------|-----|----------|-----|-----------|
| 128 | END      | 154 | CONT     | 180 | SGN       |
| 129 | FOR      | 155 | LIST     | 181 | INT       |
| 130 | NEXT     | 156 | CLR      | 182 | ABS       |
| 131 | DATA     | 157 | CMD      | 183 | USR       |
| 132 | INPUT #  | 158 | SYS      | 184 | FRE       |
| 133 | INPUT    | 159 | OPEN     | 185 | POS       |
| 134 | DIM      | 160 | CLOSE    | 186 | SQR       |
| 135 | READ     | 161 | GET      | 187 | RND       |
| 136 | LET      | 162 | NEW      | 188 | LOG       |
| 137 | GOTO     | 163 | TAB (    | 189 | EXP       |
| 138 | RUN      | 164 | ТО       | 190 | cos       |
| 139 | IF       | 165 | FN       | 191 | SIN       |
| 140 | RESTORE  | 166 | SPC      | 192 | TAN       |
| 141 | GOSUB    | 167 | THEN     | 193 | ATN       |
| 142 | RETURN   | 168 | NOT      | 194 | PEEK      |
| 143 | REM      | 169 | STEP     | 195 | LEN       |
| 144 | STOP     | 170 | +        | 196 | STR\$     |
| 145 | ON       | 171 | _        | 197 | VAL       |
| 146 | WAIT     | 172 | *        | 198 | ASD       |
| 147 | LOAD     | 173 | /        | 199 | CHR\$     |
| 148 | SAVE     | 174 | 1        | 200 | LEFT\$    |
| 149 | VERIFY   | 175 | AND      | 201 | RIGHT\$   |
| 150 | DEF      | 176 | OR       | 202 | MID\$     |
| 151 | POKE     | 177 | >        | 203 | GO        |
| 152 | PRINT#   | 178 | =        | 204 | ?SYNTAX   |
| 153 | PRINT    | 179 | <        |     | ERROR (注) |
|     |          |     |          |     |           |

(注)変換後、このエラーメッセージが出力されます。

# B:CBM BASICの省略形

以下に、キーワードの省略形の一覧表を示します。○で囲んだアルファベット □ |SHIFT| キーを押しながら、そのキーを押すことを示します。その左に、ディーノレイに表示される形を示します。 ( ) 内は片仮名モードの場合です。

| キーワード  | 省     | 略形                  | キーワード   | 省              | 略形                      |
|--------|-------|---------------------|---------|----------------|-------------------------|
| AND    | A(N)  | A∠(Aホ)              | RUN     | $R \mathbb{Q}$ | R (R1)                  |
| NOT    | NO    | N[_(N.△)            | SAVE    | s@             | S ♠ (Sチ)                |
|        |       |                     | STEP    | STE            | ST ̄(STナ)               |
| CLOSE  | CL®   | CL∏(CLマ)            | STOP    | \$®            | s ∐(sヤ)                 |
| CLR    | C©    | C∏(Cリ)              | SYS     | SŶ             | s∏(sル)                  |
| CMD    | CM    | c∑(c√)              | THEN    | $T \mathbb{H}$ | Τ[∐(Τネ)                 |
| CONT   | CO    | C[(C₹)              | VERIFY  | VE             | v 二(vナ)                 |
| DATA   | DA    | D ♠ (Dチ)            | WAIT    | WA             | <b>₩</b> ♠( <b>W</b> チ) |
| DEF    | DE    | ロー(ロナ)              |         |                |                         |
| DIM    | D①    | ロン(ロノ)              | ABS     | AB             | A ∏(Aツ)                 |
| END    | ΕN    | E (Eホ)              | ASC     | A(S)           | A ♥(Aモ)                 |
| FOR    | F@    | F□(Fマ)              | ATN     | AT             | A [(A+)                 |
| GET    | GŒ    | <b>G</b> ⊟(Gナ)      | CHR\$   | CH             | C ∏(Cネ)                 |
| GOSUB  | GO®   | GO♥(GOモ)            | EXP     | E®             | E ♣(Eリ)                 |
| GOTO   | G@    | G[□(Gマ)             | FRE     | F®             | F 🔲 (F x )              |
| INPUT# | 1 (N) | l ☑(lホ)             | LEFT\$  | LE®            | LE (LE =)               |
| LET    | L®    | に (にナ)              | MID\$   | ΜŪ             | M∑(M∠)                  |
| LIST   | L①    | ∟□(∟ノ)              | PEEK    | PE             | P (Pナ)                  |
| LOAD   | L®    | □□(□□)              | RIGHT\$ | RU             | R [ (R / )              |
| NEXT   | N®    | N-(Nナ)              | RND     | RN             | R ☑(Rホ)                 |
| OPEN   | OP    | 0□(0ミ)              | SGN     | S©             | s∏(sヌ)                  |
| POKE   | PO    | P∐(P∠)              | SIN     | S①             | s∏(s≠)                  |
| PRINT  | 7     | ?                   | SPC (   | S®             | s∏(s≷)                  |
| PRINT# | P®    | P[[(bx)             | SQR     | s@_            | S • (SA)                |
| READ   | RE    | R <sup>(</sup> (Rナ) | STR\$   | STR            | STL (STX)               |
| RE-    | RE(S) | RE♥(RE±)            | TAB (   | T(A)           | Υ ♠ (Tチ)                |
| STORE  |       |                     | USR     | U(S)           | U ♥(Uモ)                 |
| RETURN | RE(T) | RE   (RE+)          | VAL     | $V^{(A)}$      | ∨ <b>♠</b> (∨チ)         |

# C:他のBASICからCBM BASICへの変換

他のBASICで書かれたプログラムをVIC-1001パーソナル・コンピューターで実行する場合、多少プログラムをCBM BASIC用に変更する必要があります。どのようにするのかの例を次に説明します。

### C.1 文字列(ストリング)配列

文字列(ストリング)の長さを宣言するのに使用されるすべてのステートメントを消去して下さい。たとえば、DIM A\$ (I、J) は、Jの要素を持った長さ I の 文字配列を表わしますが、これはCBM BASICでは、DIM A\$ (J) に書き直して下さい。

いくつかのBASICでは、コンマ (,) やアンド (&) を文字列 (ストリング) を連結するのに使用していますが、これはCBM BASICでは、プラス符号 (+) を用いて下さい。

CBM BASICでは、MID\$、RIGHT\$、LEFT\$関数を、文字列(ストリング)の中から指定した文字数分だけとりだして、文字列を作るのに使用します。他のBASICでは、A\$(I) はA\$の中から第 I 番目の 1 文字をとりだします。またA\$(I、J) はA\$の第 I 番目から J番目の文字列をとりだします。これをCBM BASICに変換する場合は次のようにします。

他のBASIC

CBM BASIC

A \$ (I) = X \$ A \$ = LEFT \$ (A \$ , I-1) + X \$ + MID \$ (A \$ , I+1)

A\$(I,J)=X\$ A\$=LEFT\$(A\$,I-1)+X\$+MID\$(A\$,J+1)

### C.2 複数の代入文

いくつかのBASICでは次のような形を用いて、BとCに同時に多を代入することができます。

### 10 LET B=C=0

しかしCBM BASICでは2番目の等号 (=) を論理演算子として処理するために、Cが $\emptyset$ の場合、Bには-1が代入されます。

そこで、CBM BASICでBおよびCにØを代入するためには次のようにします。

### 10 C=0: B=0

#### C.3 マルチステートメント

複数のステートメントを1行上にならべて書く場合、それぞれのステートメントをバックスラッシュ(\)で区切って書くBASICがありますが、CBM BASICの場合は、コロン(:)を使用して区切ります。

#### MAT開数

MAT関数を用いている場合、CBM BASICでは、FOR NEXTループを使用して書きなおして下さい。

# D:エラーメッセージー覧表

| 原  DIMステートメントで確保された配列の範囲を越えたような場合、またDIM宣言をせずに、配列を使用した場合(ただし、BASICの文法上、配列の添え字がの~9までの範囲では、DIM宣言する必要はありません)。  列) DIM A(2,2) と宣言したあと A(1,1,1) = 2を実行すると 7BAD SUBSCRIPT ERRORになります。  DIM宣言をしないで直接 A(10,10) = 2を実行すると 7BAD SUBSCRIPT ERRORになります。  通常、STOPコマンドの実行、もしくは STOP キーを押すことにより、プログラムの実行を中止させた場合に、CONTによりSTOPを解除して、引き続き実行を再開できま |
|---|
| うな場合、またDIM宣言をせずに、配列を使用した場合(ただし、BASICの文法上、配列の添え字がの~9までの範囲では、DIM宣言する必要はありません)。  列) DIM A(2,2) と宣言したあと A(1,1,1) = 2を実行すると ?BAD SUBSCRIPT ERRORになります。  DIM宣言をしないで直接 A(10,10) = 2を実行すると ?BAD SUBSCRIPT ERRORになります。  通常、STOPコマンドの実行、もしくは STOP キーを押すことにより、プログラムの実行を中止させた場合に、   |
| A(1,1,1) = 2を実行すると ?BAD SUBSCRIPT ERRORになります。  DIM宣言をしないで直接 A(10,10) = 2 を実行すると ?BAD SUBSCRIPT ERRORになります。  通常、STOPコマンドの実行、もしくは STOP キーを 押すことにより、プログラムの実行を中止させた場合に、   |
| A (10,10) = 2 を実行すると<br>?BAD SUBSORIPT ERRORになります。<br>通常、STOPコマンドの実行、もしくは STOP キーを<br>押すことにより、プログラムの実行を中止させた場合に、  |
| <b>事すことにより、プログラムの実行を中止させた場合に、</b>   |
| tが、次の4通りの場合に、CONTを実行させようとする<br>と、このエラーが発生します。   |
| )プログラムが存在しない場合  |
| 2)STOPした後に、新しい行を追加した場合  |
| 3) プログラムがRUNコマンドにより実行されていない<br>場合   |
| 4)エラーが発生して実行が中断された場合  |
| 除算(割り算)において分母をので実行しようとした場合に、このエラーがおきます。<br>もしプログラム実行中に、このエラーメッセージが出力されたなら、次の方法で、分母がのであることを、確めることができます。  |
| 1   |

| エラーメッセージ               | 原   |
|------------------------|---|
| DIVISION BY ZERO       | ? DIVISION BY ZERO ERROR IN 10       LIST 10     (エラーの発生した行のリストを調べます)       10 A=B/C     (このような計算式だったとします)       PRINT C     (ダイレクトモードで分母Cの値を調べますと、のになってることがわかります)  |
| FORMULA TOO<br>COMPLEX | 文字列式(ストリング式)を評価するさいに、文字列式が<br>複雑すぎて、メモリー内の文字列のポインターの領域をこ<br>えてしまった場合に発生します。<br>文字列式をいくつかにわけて短かくすることにより、解<br>決することができます。   |
| ILLEGAL DIRECT         | BASICで、INPUTまたはGETでデータを入力する場合、88文字(=88バイト)長の入力バッファーを使用します。この同じバッファーが、ダイレクトモードで入力されたコマンドをとりこんで実行させるためにも使用されます。そのため、INPUTおよびGETを、ダイレクトモードで使用すると、このエラーが発生します。 DEFもまた、多少ちがいますが、同じように、バッファーの関係でダイレクトモードで使うことができません。つまり関数名は、関数定義をおこなった式のポインターといっしょにBASICの変数エリアに格納されます。そのため関数は、入力バッファの中にだけ残されるため、次のコマンドが何か入力されると、関数がなくなってしまいます。DEFをダイレクトモードで使用しますとこのエラーが発生します。 |
| ILLEGAL QUANTITY       | 関数に、規定にあわない数値や変数を用いて実行しようとした場合、このエラーが発生します。  1) 配列の添字が次の範囲外の場合  Ø≤(添字) ≤32767  たとえばA(-5) = 5 は  ?ILLEGAL QUNTITY ERRORになります。  2) LOG(X) (X>Ø)  Xが負またはØの場合エラーになります。  |

| エラーメッセージ        | 原    因  |
|-----------------|---|
|                 | <ul><li>3) SQR (×) (×≥∅)</li><li>Xが負の場合、エラーになります。</li></ul>     |
|                 | 4) A↑B<br>Aが負で、かつBが整数でない場合、エラーになります。                            |
|                 | PRINT (-5) †2.3<br>?ILLEGAL QUANTITY ERROR                      |
|                 | 5) 機械語ルーチンをセットしないでUSR関数を使用するとエラーになります。                          |
|                 | 6)MID\$,RIGHT\$,LEF\$の文字列の長さを指定するパラメータの数値が下記の範囲外の場合、エラーになります。   |
|                 | 1 ≤ (パラメーター) ≤255   |
|                 | 7) ON 〈式〉GOTO (2.20参照) のインデックス (式の評価<br>結果) が次の範囲外の場合、エラーになります。 |
|                 | Ø ≤ (インデックス) ≤ 255  |
|                 | 8) PEEK, POKE, WAIT, SYSで指定するアドレスが次の<br>範囲外の場合、エラーになります。        |
|                 | Ø ≤アドレス ≤65535  |
| -               | 9) WAIT, POKE, TAB, SPCで使用するパラメータが次の<br>範囲外の場合、エラーになります。        |
|                 | ∅ ≤ (パラメータ) ≤255  |
|                 | たとえば、POKE826、1000を実行すると<br>?ILLEGAL QUANTITY ERROR<br>になります。    |
| NEX WITHOUT FOR | FORNEXTのネスティングが正しくプログラムされな<br>かったか、または、NEXTに対応するFORがない場合に、      |

| エラーメッセージ        | 原  |
|-----------------|--|
| NEX WITHOUT FOR | このエラーが発生します。   |
|                 | 例) FOR I=1T010:NEXT:NEXT ?NEXT WITHOUT FOR ERROR  FOR I=1T010:NEXT J ?NEXT WITHOUT FOR ERROR   |
| OUT OF DATA     | R E A D を実行して読みこもうとするデータの数が、 D .<br>A T A で定義されているデータの数より多い場合に、エラーが発生します。  |
|                 | また、ディスプレイの画面に、READY.の文字が表示されている同じ行にカーソルを移動し、RETURN (リターン) キーを押すとこのエラーが発生します(なぜならREADY.をBASICがREADY.という命令として実行しょうとするからです)。  |
| OUT OF MEMORY   | プログラムを作成または編集している場合、BASICテキストエリア(メモリーマップ参照)がいっぱいになった場合、エラーになります。 プログラム実行時には、変数を割りつけたり、作られたりすることにより、メモリーがいっぱいになった場合、エラーになります。 プログラムが短かくても、DIMにより配列宣言をすると、大きなメモリー領域を確保しますので注意して下さい。また、FORNEXT、GOSUBRETURNのように、スタックを使用する場合(2.9,2.11参照)、スタックの容量をこえた場合、メモリーに関係なく、このエラーが発生します。たとえば  10 GOSUB10: GOTO10 RUN |

| エラーメッセージ        | 原因   |
|-----------------|--|
|                 | 実際にメモリー容量が足りなくなったのか、単にスタックの容量が一杯になってエラーになったのかは FR E関数を用いて残りのメモリー容量を調べて判断することができます。   |
|                 | PRINT FRE (Ø)  |
| OVERFLOW        | 計算結果や入力した数値が1.70141184E+38より大きい場合、BASICでは扱うことができず、エラーになります。なお、UNDERFLOWはエラーとされませんが、2.93873587E-39より小さい数値は、すべてのとして扱われます。 PRINT 1E40 ?OVERFLOW ERROR |
| REDIM'D ARRAY   | <ul><li>一度宣言した配列を同じ変数名でもう一度宣言しなおすとエラーになります。</li><li>例) A(5)=6<br/>DIM A(10,10)<br/>? REDIM'D ARRAY ERROR</li></ul>                                 |
| REDO FROM START | これは他のエラーとちがって、プログラムの実行を中止することはありません。 INPUT実行時に、INPUTで指定したのとちがう型のデータ、(たとえば数値 =ニューメリック を指定しているのにアルファベットを入力した場合) を入力した場合に発生します。                       |
|                 | 例)10 INPUT A RUN ?ABC (*ABC"を入力します) ?REDO FROM START ? (再び入力を要求します) 正しくデータが入力されるまでによる入力要求がくり返されます。   |

| エラーメッセージ                | 原    因   |
|-------------------------|--|
| RETURN WITHOUT<br>GOSUB | 対応するGOSUBがなしで、RETURNを実行しようとした場合、エラーになります。  |
|                         | 例)CLR (2.2参照)<br>?RETURN WITHOUT GOSUB ERROR   |
| STRING TOO LONG         | 255文字より長い文字列(ストリング)を作ろうとした場合に、このエラーが発生します。   |
|                         | 例)A\$="A":FORI=1T010:A\$=A\$+A\$:NEXT<br>?STRING TOO LONG ERROR  |
| SYNTAX                  | BASICの文法上の誤り、つまり、コマンド、ステートメント、関数として、CBM BASICが定めていない形式、文字を使用した場合、文字のつづりをまちがえた場合に、このエラーが発生します。また、かっこの数があわない場合にも、エラーになります。 |
|                         | 例) A=(1+8<br>?SYNTAX ERROR   |
| TYPE MISMATCH           | 代入文などで式の左右の形が一致していない場合に、こ<br>のエラーになります。  |
|                         | 例)A\$=12<br>?TYPE MISMATCH ERROR   |
| UNDEF'D FUNCTION        | 定義していない関数を使用しようとした場合、エラーが<br>発生します。  |
|                         | 例)X=FNA(3)<br>?UNDEFD FUNCTION ERROR   |
| UNDEF'D<br>STATEMENT    | GOTO, GOSUB、THENで指定したライン・ナンバーが<br>プログラム中にない場合、エラーが発生します。   |
|                         | 例) GOTO A<br>7 UNDEF'D STATEMENT ERROR   |

| オペーレ           | ·ーティング・システム(OS)エラー                     |
|----------------|--|
| エラーメッセージ       | 原    因                                 |
| BAD DATA       | 指定した機器(デバイス)よりデータを入力した場合、              |
|                | 数値(ニューメリック)データであるべきなのに文字(ア             |
|                | ルファ)データが入力された時に、このエラーが発生しま             |
|                | す。つまりファイル上にあるデータの形式がまちがってい             |
|                | る場合に、発生します。                            |
| DEVICE NOT     | シリアルバスにつながっていない機器(デバイス)を使              |
| PRESENT        | 用しようとした場合に、発生します。この時、ST(ステ             |
|                | ータス、3.21参照) はタイムアウトを示す2になります。          |
|                | OPEN, CLOSE, CMD, INPUT#, GET#, PRINT# |
|                | を実行した時に、このエラーが調べられます。                  |
| FILE NOT FOUND | OPENまたはLOADコマンドで指定されたファイル名が、           |
|                | この時いっしょに指定された機器(デバイス)から、発見             |
|                | することができなかった場合、エラーになります。                |
| FILE NOT OPEN  | ファイルをOPENしないで使用した場合に、エラーにな             |
|                | ります。                                   |
|                | <br>  例)CLR                            |
|                | INPUT#10, A                            |
|                | ?FILE NOT OPEN ERROR                   |
| FILE OPEN      | 同じファイルを2度OPENしょうとした場合、このエラ             |
|                | ーが発生します。                               |
|                | 例) OPEN1,4,1                           |
|                | OPEN1,4,1                              |
|                | ?FILE OPEN ERROR                       |
| LOAD           | カセットテーブからプログラムを読みこんでいる(ロー              |
|                | ディング)時に発生します。これはテープの最初のブロッ             |
|                | クに31個以上のエラーがあったか、または、2つのブロッ            |
|                | クの同じ位置にエラーが発見された場合に、発生します。             |
|                |  |
|                |  |

| エラーメッセージ        | 原    因   |
|-----------------|--|
| NOT INPUT FILE  | カセットテープの場合、書きこみ(ライト)用にOPEN<br>されたファイルからデータを読みこもう (リード) しよう<br>とした場合に、このエラーが発生します。ただし、書きこ<br>み用にOPENしたファイルをCLOSEして、その後、 読<br>みこみ(リード)用にOPENした場合は、エラーにはなりま<br>せん。<br>例) 10 OPEN1,1,1<br>20 INPUT #1,A<br>RUN<br>:<br>: ?NOT INPUT FILE ERROR                                       |
| NOT OUTPUT FILE | カセットテープの場合、ファイルからデータを読みこむのと同時に、書きこむことができません。そのため読みこみ (リード) 用にOPENされたファイルに、書きこみ (ライト) をおこなうと、このエラーが発生します。またキーボードは入力 (読みこみ) のみをおこなう機器 (デバイス)ですから、キーボードになにかを出力 (書きこみ) することはできません。そこで、次のような場合にはエラーになります。  例) 10 OPEN1,0 (キーボードのデバイス番号=0) 20 PRINT #1,A\$ RUN ? NOT OUTPUT FILE ERROR |
| VERIFY          | メモリーに格納されているプログラムの内容と、指定されたファイルの内容がちがっている場合、このエラーが発生します。   |

# E:誘導関数

CBM BASICが組み込み関数として用意していない関数のうち、いくつかは組 み込み関数を用いて作ることができます。以下の数式を参考にして下さい。(誤 差の範囲に注意が必要です)。

| <b>関数名</b>                 | 計算方法  |
|----------------------------|---|
| SECANT                     | SEC(X) = 1/COS(X)                                 |
| COSECANT                   | CSC(X) = 1/SIN(X)                                 |
| COTANGENT                  | COT(X) = 1/TAN(X)                                 |
| INVERSE SINE               | ARCSIN(X) = ATN(X/SQR(-X * X+))                   |
| INVERSE COSINE             | ARCCOS(X) = $-ATN$<br>$(X/SQR(-X * X+1)) + \pi/2$ |
| INVERSE SECANT             | ARCSEC(X) = ATN(X/SQR(X * X - 1))                 |
| INVERSE COSECANT           | ARCCSC(x) = ATN(x/SQR(x * x-1))                   |
|                            | $+(SGN(X)-1) * \pi/2$                             |
| INVERSE COTANGENT          | $ARCOT(X) = ATN(X) + \pi/2$                       |
| HYPERBOLIC SINE            | SINE $(X) = (EXP(X) - EXP(-X))/2$                 |
| HYPERBOLIC COSINE          | COSH(X) = ((EXP(X) + EXP(-X))/2                   |
| HYPERBOLIC TANGENT         | TANH(X) = EXP(-X)/EXP(X)                          |
|                            | +EXP(-X)) *2+1                                    |
| HYPERBOLIC SECANT          | SECH(X) = 2/(EXP(X) + EXP(-X))                    |
| HYPERBOLIC COSECANT        | CSCH(X) = 2/(EXP(X) - EXP(-X))                    |
| HYBERBOLIC COTANGENT       | COTH(X) = (EXP(-X)/(EXP(X))                       |
|                            | -EXP(-X)) *2+1                                    |
| INVERSE HYPERBOLID SINE    | ARCOSINH(X) = LOG(X + SQR(X * X + 1))             |
| INVERSE HYPERBOLIC         | ARCCOSH(X) = LOG(X + SQR(X * X - 1))              |
| COSINE                     | ADCTABUL(V) -1 00 (/1 1 V) //1 V) //1             |
| INVERSE HYPERBOLIC TANGENT | ARCTANH(X) = LOG((1+X)/(1-X))/2)                  |
| INVERSE HYPERBOLIC         | ARCSECH(X)=LOG((SQR(-X * X+1)<br>+1) /X))         |
| SECANT                     | ARCCSCH(X) = LOG((SGN(X))                         |
| INVERSE HYPERBOLIC         | *SQR(X *X+1)/X))                                  |
| COSECANT                   | ARCCOTH(X) = LOG((X+1)/(X-1))/2                   |
| INVERSE HYPERBOLIC         | $ARCCOTT(A) = LOG((A \pm 1)/(A \pm 1))/2$         |

COTANGENT

# F: カラーコントロール

VIC1001でカラーをコントロールするのに幾通りかの方法があります。

バックカラーは、スクリーンカラーとボーダーカラーを同じように、または、 各自別々にコントロールすることができます。

キャラクターカラーのコントロールは、CTRL キーを使うことにより、また PRINTあるいはPOKEによりそれぞれの方法でおこなうことができます。

#### F.1. バックカラーのコントロール

パックカラーのコントロールをおこなうためには、メモリー中の(900F) $_{16}$  帯地 ( $=(36879)_{10}$ 番地)にコントロール用のデータを書きこみます。書きこむデータは $\emptyset$ から255の範囲の数値です( $\emptyset \le (データ) \le 255$ )。

では実際にバックカラーをコントロールするプログラム例をあげましよう。

- 例) 10 R=INT(RND(1) \*255) +1……書きこむデータを作ります。
  - 20 C=36879

·····(900F)16番地

- 30 POKEC, R
- 40 FORI=1TO2000: NEXT ……次の色へ移るまでの時間を調節します。
- 50 GOTO 10

RUN

この場合、Rは1から255までの範囲内の数値をランダムに36879番地にかきこみ次々と色をかえていきます。この時、キャラクターカラーは変わらず、また、Rの値を2進法であらわした場合、 $2^3$ ビットがON (オン)の場合、画面の文字はリパース(反転)になって出力されます。たとえば、R=139の場合、Rを2進数で表現すると、R=199091911となり $2^3$ ビットが1(=ON)なので、この場合はリパースで出力されます。

次にもうひとつカラーをコントロールするプログラム例をあげて説明しましよう。

(91) 10 C= 9 × 16 1 3 + 15

20 X=INT (RND(1) \* 15)

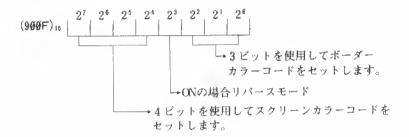
36 Y=INT(RND(1) \* 7)

40 R = X \* 16 + 8 + Y

- ··· (900F)16 番地
- …スクリーンカラーコードを作ります
- …ボーダーカラーコードを作ります。
- …スクリーン、ボーダーカラーコードをそれ ぞれの位置にセット、8はリバースにする ため

5g/ POKEC, R …カラーコードを書きこみます。 6g/ FOR I=1 TO2gg/: NEXT …次の色へ移るまでの時間を調節します。 7g/ GO TO 2g/ RUN

この場合、スクリーンとボーダーカラーが各々、変化していきます。 ライン・ナンバー40についてもう少し説明しますと、カラーコントロールする りめに、(900F) 16番地の1バイト (=8ビット) を使用しますが、このバイトの各 ビットは次のような意味をもちます。



| 〈スクリ | ーンカラーコード〉 | 〈ボーダーカラーコード〉 |
|------|-----------|--------------|
| コード  | 色         | コード 色        |
| Ø    | ブラック      | ∅ ブラック       |
| 1    | ホワイト      | 1 ホワイト       |
| 2    | レッド       | 2 レッド        |
| 3    | シアン       | 3 シアン        |
| 4    | マジェンタ     | 4 マジェンタ      |
| 5    | グリーン      | 5 グリーン       |
| 6    | ブルー       | 6 ブルー        |
| 7    | イエロー      | 7 イエロー       |
| 8    | オレンジ      |              |
| 9    | ライト オレンジ  |              |
| 10   | ピンク       |              |
| 11   | ライト シアン   |              |
| 12   | ライト マジェンタ |              |
| 13   |           |              |
| 14   | ライト ブルー   |              |
| 15   | ライト イエロー  |              |

次のプログラムによりスクリーンカラーがコードによりどのような色になるかたしかめて下さい。

10 INPUTA

20 POKE36879. A \* 16+8

30 GOTO10

RUN

?……→ Ø から15までのコードをキーボードから入力して下さい

スクリーンカラーコードが8以上になると明るい色になるため、キャラクターカラーに濃い色を使用すると、非常に鮮明な、見やすい画面を作成することができます。

#### F.2 キャラクターカラーのコントロール

キャラクターカラーのコントロールの方法には次の3通りの方法があります。

1) キーボードからコントロールキーと数字を組みあわせて同時に入力すること によりコントロールできます。 たとえば

CTRL キー十丁とおすとブラックになります。

CTRL キー十7とおすとブルーになります。

数字は1~8まで使用できます(第1部ハード編第4章第3節を参照)。

カラー用のコントロール文字をPRINTで出力することによりコントロールすることができます。

例をあげますと、

10 C\$ = " ■= 15 16 KERT "

…コントロール文字

20 C=INT(RND(1) \*B)+1

…カラー指定のために1~8のランダムな 数を作ります。

30 PRINTMID\$(C\$,C,1); "A"; …Aの文字を色々なカラーで出力します。

40 GOTO20

(®Aが表示されていない部分はパックカラー) と同じ色のためみえないからです。 カナモードの場合は CS は以下のようになります。

- 10 C\$ = " UNESDIGHT"
- 20 C=INT(RND(1) \*B)+1
- 30 PRINTMID\$ (C\$,C,1): " A":
- 40 GOTO20

RUN

ライン・ナンバー10でカラーのコントロール文字を作るには、まず、クォーテーション (\*) をキーボードから入力したあと、 CTRL キーを押しながら、数字の 1 から 8 まで順番に押していきます。

- 3) スクリーンに直接POKEを用いて文字を書いたり、かつカラーをコントロールすることができます。カラーコントロールを説明する前にまず、画面に文字を出力するために用いるスクリーンコードを、ASCII (アスキー) コードから作りだすプログラムを書いてみましよう。
  - 5 INPUTX%
  - 10 W%=X%AND63
  - 20 IF (X%AND128) < >0THENW %=W %OR64
  - 30 X % = W %
  - 40 PRINTX %

最初にX%にスクリーンコードに変換したいASCIIコード、たとえば文字Aに対応する (65)nを代入して、このプログラムを実行すると、結果のスクリーンコードX% (ライン・ナンバー30) には1が代入されます (第1部第4章第3節のキャラクター・コード表中のスクリーンコードを参照して下さい)。画面に文字をサパースにして出力させる場合は、スクリーンコードに128を加算します。

両面(スクリーン)は、506文字分のエリアを持っています。メモリー内のどこ □あるのかは、次のようにメモリーの容量により2通りあります。

- 1) RAMエリアが(1999)<sub>16</sub>~(1FFF)<sub>16</sub>番地および3K 拡張メモリーを装備して、(9499)<sub>16</sub>~(1FFF)<sub>16</sub>番地の場合、スクリーンは(1E99)<sub>16</sub>~(1FFF)<sub>16</sub>番地にあります。
- 2) RAMエリアが(1例例)<sub>18</sub>から(2例例)<sub>18</sub>番地以降最大(7FFF)<sub>16</sub>番地まである場合、 (この時(例4例)<sub>16</sub>~(例FFF)<sub>16</sub>番地のRAMの有無には無関係)、スクリーンは、 (1例の)<sub>16</sub>~(11FF)<sub>16</sub>番地にあります。

さて、この5%6文字の各々についてのカラーの指定ができます。スクリーンのエリア5%6文字に対応して(96%%) $_{16}$ ~(97FF) $_{16}$ 番地までが、それぞれのカラーのコントロールをおこないます。カラーコントロールコードは、ボーダーカラーコード(D.1参照)と同じです。

では、画面いっぱいに真赤なハートを書いていくプログラムを作ってみましよう。

10 SM=7680 (またはSM=4096)…前述の①、②を参照して下さい。

20 SC=38400

30 FOR!=0TO505

40 M=SM+I:C=SC+I

50 POKEM. 83 : POKEC. 2

60 NEXT

RUN

ライン・ナンバー10でスクリーンエリアのスタート番地は、前述の①、②の2 通りありますから、自分の機械にあわせて選択して下さい。

ライン・ナンバー20のカラー指定のためのエリアのスタート番地は、スクリーンエリアの場所が変わっても、常に一定の38400番地からはじまります。

# G: サウンド

カラーをコントロールするのと同じように、ある決められた場所に数値をPOKE より代入することにより、サウンド(音)を作ることができます。決められた 当所をサウンド(音)をコントロールするレジスターといいます。レジスターは、  $_{1}$  間あり、各々に、(900 A)  $_{16}$ 、(900 B)  $_{16}$ 、(900 C)  $_{16}$ 、(900 D)  $_{16}$ 、(900 E)  $_{16}$  番  $_{1}$  かのアドレスが割りあてられています。各レジスターに、次のような役割があり しょ

レジスターアドレス〉 〈機能〉

(900A)<sub>16</sub>番地 コードにより周波数の異なった音を出します(音の高低

がつくれます)。コードはサウンドコード表を参照して下

さい。

(900B)<sub>16</sub>番地 機能は(900A)<sub>16</sub>番地と同じですが、音色(トーン)がちが

っています。

(900C)<sub>16</sub>番地 機能は上の2つと同じですが、音色(トーン)がちがって

います。

(900D)<sub>16</sub>番地 ノイズを作ります。

(900E)<sub>16</sub>番地 コードにより音の振幅を決定します。(音の大小を作りま

す) この場合のコードは Ø から15までの範囲で15が一番

大きい音をだします。

音の高低はサウンドコード (128 $\leq$ (コード) $\leq$ 255) により指定することができ 上す。コードの値が大きくなれば、音も高くなります。ただし例外として255の場 古は、低くなってしまいます。また $\emptyset$ をレジスターに代入すると音が消えます。

# 〈サウンドコードー覧表〉

| 音階                                 | コード | 音階                                | コード |
|------------------------------------|-----|-----------------------------------|-----|
| ド (C)                              | 128 | ソ (G)                             | 213 |
| ド# (C#)                            | 134 | ソ* (G *)                          | 215 |
| レ (D)                              | 141 | ラ (A)                             | 217 |
| レ* (ロ#)                            | 147 | ラ# (A #)                          | 219 |
| ≷ (E)                              | 153 | シ (B)                             | 221 |
| ファ (F)                             | 159 | F (C)                             | 223 |
| ファ <sup>‡</sup> ( F <sup>#</sup> ) | 164 | K* (C*)                           | 225 |
| ソ (G)                              | 170 | レ (D)                             | 227 |
| ソ# (G#)                            | 174 | レ* (D *)                          | 228 |
| ラ (A)                              | 179 | ₹ (E)                             | 230 |
| ラ* (A *)                           | 183 | ファ (F)                            | 231 |
| シ (B)                              | 187 | ファ <sup>‡</sup> (F <sup>‡</sup> ) | 232 |
| К (C)                              | 191 | ソ (G)                             | 234 |
| К# (С#)                            | 195 | ソ# (G#)                           | 235 |
| レ (D)                              | 198 | ラ (A)                             | 236 |
| レ <sup>‡</sup> (D <sup>#</sup> )   | 201 | ラ* (A *)                          | 237 |
| ₹ (E)                              | 204 | シ (B)                             | 238 |
| ファ (F)                             | 207 | К (C)                             | 239 |
| ファ <sup>#</sup> (F #)              | 210 | F# (C#)                           | 240 |

(注) この一覧表の音階は絶対音階ではありません。

# H: ハイ・レゾリューション

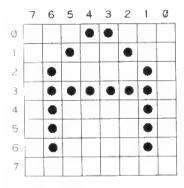
ハイ・レゾリューションとは高分解能ということです。この機能を利用すると、 音符の \*」" や、 \*木"、\*■" などのキーボードにない漢字や、グラフィックな どを自由に表示することができます。

自分の好きな文字やグラフィックを作るのに、次の3通りの方法があります。

- ○8×8ドット構成
- ○8×16ドット構成
- ○4×8ドット、または4×16ドット構成でマルチカラーコントロールをおこなう。

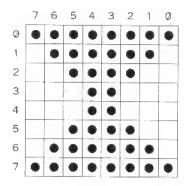
まず、 $8\times8$ ドットとか $8\times16$ ドットというのは、具体的にどういうことなのか説明じましょう。

画面に表示される文字は、各々、横に8 個、縦に8 個の点(ドット)で構成されており、それらの点をON(オン)、OFF(オフ)、すなわち、明るくしたり、暗くしたりすることにより作られています。だとえば、 $^{8}A''$  という文字は、



●…ON(オン) □…OFF(オフ)

という形でつくられています。では、もし〝द्य″というグラフィック文字を8× 8ドット構成で作るとどうなるかみてみましょう。



では、こうした方法で、8×8または8×16ドットの好きなパターンを作り、 それを画面に表示する方法について説明しましょう。

まず、表示するパターンが $8\times8$ ドットか $8\times16$ ドットかを選択するために、 $(9003)_{16}$ 番地( $=(36867)_{19}$ 番地)の $2^9$ ビットをON(オン)かOFF(オフ)つまり1かりにしなければなりません。この場合、 $2^9$ がりの時は $8\times8$ ドット構成、1の時は $8\times16$ ドット構成であることを示します。

# 例) (9003):6番地=(36867):6番地の2<sup>9</sup>ビットが=00場合…8×8ドット構成 = 1の場合…8×16ドット構成

次に指定したドット構成で作成した文字パターンコードを格納する場所を指定します。

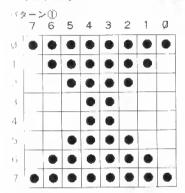
たとえば、 $(\emptyset400)_{16}$ 番地からを使用する場合は、 $(9005)_{16}$ 番地 (= $(36869)_{10}$ 番地) の下4ビット  $(2^{6}, 2^{1}, 2^{2}, 2^{3}$ ビット) に次のようにセットします。



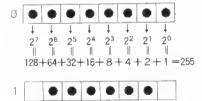
また、(3000)  $_{16}$  番地からを文字パターンコードの格納番地として使用する場合は、次のようになります。

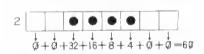


では、文字パターンコードを実際につくってみましょう。

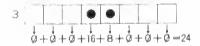


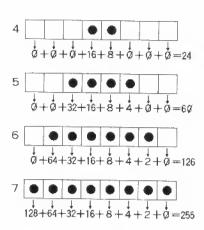
このような場合、左端から右横へ8ドットの各ドットに値をもたせ、その合計を1パイトのデータとして、8行分、8パイトのコードを作ります。つまり●でぬりつぶしてあるところに、それに対応する値を代入して、その合計をつくります。パターン①の場合、上の行から





0 + 64 + 32 + 16 + 8 + 4 + 2 + 0 = 126

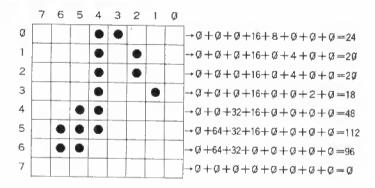




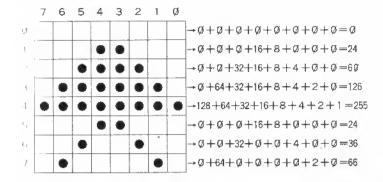
各行( $\emptyset$ ~7)の右側に書かれている合計値が文字パターンコードになります。  $\blacksquare$ を表示するためには、この8個のそれぞれの合計値(8バイト使用します)が必要になります。

文字パターンコードの作り方の例をいくつかあげてみましょう。

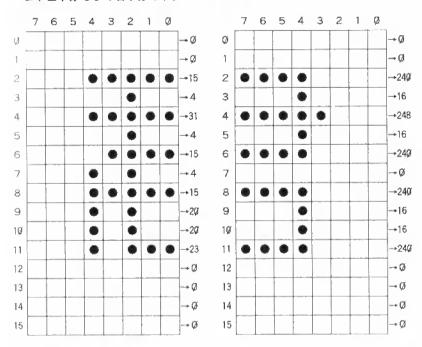
#### 例1) 音符♪の場合



#### 例 2 ) : ♠ "の場合



例3) 8×16ドット構成の文字2つを使って漢字の〝君″を作ってみました。 まず左半分そして右半分です。



8×16ドット構成の場合は、1つの文字を作るために16パイト必要になります。

以上の例1.2.3、で文字パターンコードの作り方がわかったところで、これらを 指定した格納番地に格納しましょう。

文字は8×8ドット構成で、格納スタート番地は(1409)ie番地とします。格納する文字パターンコードは、例1、例2で作成したものを使用します。

①8×8ドット構成であることを指定します。

#### POKE36867, PEEK (36867) AND 254

これで $(99993)_{16}$ 番地 $(=(36867)_{19}$ 番地)の $2^9$ ピットを $\emptyset$ にして $8 \times 8$ ドット構成であることが指定されます。なお $8 \times 16$ ドット構成の指定はPOKE 36867、PEEK(36867)OR1でおこないます。

②格納スタート番地が (1400)<sub>16</sub>番地、ビデオRAMのスタート番地が (1E000)<sub>16</sub>番地であることを指定します。

POKE36869,253 POKE36866, PEEK (36866) OR128

- 注) 第3部のMPC 6560ビデオ・インターフェース・チップの項を参照して下さい。
- ③文字パターンを格納します。

例1の文字パターンを (1499):6番地から8パイト分格納します。

POKE5120,24 POKE5121,20

POKE5122.20

POKE5123.18

POKE5124,48

POKE5125,112

POKE5126,96

POKE5127, Ø

次に例2の文字パターンをつづいて格納します。

POKE5128, Ø POKE5129, 24 POKE513Ø, 6Ø POKE5131, 126 POKE5132, 255 POKE5133, 24 POKE5134, 36 POKE5135, 66

- 4)次にこれらの文字パターンを画面に表示するには、キーボードの@ (アットマーク) を押して下さい。すると例1の文字パターンが画面にあらわれます。次に「Aを押して下さい。例2の文字パターンが表示されます。格納した文字パターンは、格納順にスクリーンコードと(スクリーンコード表参照)対応します。最大512種類までの文字パターンを作成することができます。
- 5)①~③までを実際にプログラムにつくってみましょう。
  - 1 POKE51, Ø:POKE52, 19 BASICプログラムのテキストエリアは、
  - 2 POKE55, Ø:POKE56, 19 (13ØØ)18番地まであることを宣言しています。
  - 3 CLR
  - 5 FORI=1TO16
  - 10 READ A
  - 20 POKE1023+4096+1.A
  - 30 NEXT
  - 40 GOTO100
  - 50 DATA24,20,20,18,48,112,96,0
  - 60 DATA0.24.60.126.255.24.36.66
- 100 POKE36867, PEEK (36867) AND 254
  - 105 POKE36866, PEEK (36866) OR128
  - 110 POKE36869, 253
  - 149 END

プログラムをうちおわったら、まちがっていないかたしかめて下さい。

RUN RETURN と入力して下さい。

[6] キーをおすと面画に♪があらわれます。

次に「A キーをおすと♠があらわれます。

このプログラムでは、2種類のパターンしか作っていませんが、これを参考にして、たくさんのあなただけのパターンを作ってみて下さい。

なお、このプログラムを終わらせるのには RUN STOP キー+ RESTORE キー 集 即して下さい。

# 第3部



# コンポーネント・データカタログ

- 1. MPS6500マイクロプロセッサー
- 2. MPS656Øビデオ・インターフェイス・チップ (VIC)
- 3. MPS6522汎用インターフェイスアダプタ
- 4. MPS2364スタティックROM

# MPS-6500マイクロプロセッサー

# 概説

MPS 6500シリーズのマイクロプロセッサーは、ソフトウェア・コンパチズルな / イクロプロセッサー・ファミリーの最初のものです。この製品のファミリーは、ノフトウェア・コンパチブルで、アドレス可能なメモリー範囲、割込入力、クロ / ク発振チップおよびドライブの選択が可能です。MPS 6500グループのマイクロ / ロセッサーはすべて、グループ内でソフトウェア・コンパチブルで、M6800製品とバス・コンパチブルです。

ファミリー内の5つのマイクロプロセッサーは、クロック発振とドライブを持ち、4つのマイクロ・プロセッサーは、外部クロックによりドライブされます。 内部クロックは高いバーフォーマジスと低コストを目的とし、タイム・ベースに ングル・フェーズで、クリスタルまたはRC人力を用いています。外部クロック は、マルチ・プロセッサー・システムに適用され、タイミングの制御が必要な場 台川います。マイクロ・プロセッサーはいずれのバーションでも最大周波数が1 MHzか2MHzが可能です(\*A″は製品番号の添字です)。

# 特長

- 5 V単一電源
- Nチャンネル・シリコン・デプレッションロード
- 8ビット並列処理
- -56インストラクション
- 16進および2進演算
- :13アドレス・モード
- ○プログラマブル・スタック・ポインタおよび可変長スタック
- ○任意のメモリー使用可能
- 1 または 2 MHzで動作
- コパイプライン構造

# 信号の説明

#### クロック (タ1、タ2)

MPS651Xは、Vcc電圧レベルの非重複2相クロックが必要です。 MPS651Xのクロックは、内部クロック・ジェネレータにより与えられます。 クロックの周波数は、外部から制御できます。

#### アドレスバス (AB)(-AB15)

出力は、TTLコンパチブルで、標準TTL1個と13分Fの負荷をドライブできます。

#### データバス (DBØ-DB7)

データバス用に8ピンあります。双方向性で、マイクロプロセッサーからデバイス、ペリフェラルへ、またはその逆のデータ転送をおこないます。出力はスリーステートバッファで、標準TTL1個と13的pFの負荷をドライブできます。

#### データバス・イネーブル (DBE)

TTLコンパチブル入力であり、スリーステートデータの出力パッファを制御し、高レベルで、バスドライバーを動作可能にします。通常は、DBEは、チュクロックでドライブされるので、マイクロプロセッサーは兔の間だけ入力が可能です。リードサイクルの間、データバスドライバは内部的にディスエーブルの状態であり、実質的には、オープン回路となります。外部からデータバスドライバをディスエーブルにするには<math>DBEは、低レベルに保持する必要があります。

#### レディ (RDY)

入力信号であり、書込みサイクル以外のすべてのサイクルで、マイクロプロセッサーを単一サイクルにします。外の間に低レベルに変化すると、現在フェッチしているカレント・アドレスをアドレスラインに出力し、ホルト状態にします。この状態は、その後の № まで続き、レディ信号は低レベルを保ちます。したがって、マイクロプロセッサーは、最高2サイクルのダイレクト・メモリ・アクセス(DMA)と同じように、低スピードのPROMとのインタフェイスも可能です。書込みサイクルの間、レディー信号が低レベルの場合は、次の読込み動作まで、書込みは無視されます。

## 「ンタラブト・リクエスト (IRQ)

TTLコンパチブル信号であり、人力されるとマイクロプロセッサー内部で、インタラプト・シークエンスが開始します。マイクロプロセッサーは割込の人る前に実行していた命令を完結させます。同時に、ステータス・コード・レジスタの割込マスクビットをチェックします。割込マスクフラグがセットされていないなら、マイクロプロセッサーは、インタラプト・シークエンスを実行しはじめます。プログラムカウンタ、ステータス・レジスタは、スタックに待避されます。次にマイクロプロセッサーは、割込マスクフラグをハイにして、それ以外のインタラプトには応じません。その後、プログラム・カウンタの下位をFFFE番地から、上位をFFFF番地からロードして、そのアドレスに制御を移します。RDY信号は、インタラブトの受けつけ時には高レベルでなくてはなりません。ワイヤOR操作には3K $\Omega$ の外部抵抗を用います。

#### ノンマスカブル・インタラプト (NMI)

この立下がり入力により、マイクロプロセッサー内部で、ノンマスカブル・インタラプト・シークエンスが発生します。

NMIは無条件削込みです。現在実行中の命令を完了し、割込マスクフラグの状態に関係なくTRQの動作シークエンスが開始します。プログラム・カウンタにロードされるアドレスは、下位、上位、それぞれFFFA番地、FFFB番地にあり、そのアドレスにプログラム制御を移します。そこにロードしてある命令により、メモリー内のノンマスカブル・インタラプト・ルーチンにジャンプします。ワイヤOR操作には、Vecに対して3kℓの外部抵抗を用います。

IRQおよびNMI人力は、系でのハードウェア・インタラブト・ラインであり、現在実行中の命令を完了してから、系の立下がりにより検出されます。

# セット・オーバフロー・フラグ (SO)

この立下がり入力により、ステータス・コード・レジスタのオーパフロービットをセットします。この信号はずの立下がりにより検出されます。

#### SYNC

出力ラインであり、マイクロプロセッサーがOP CODEのフェッチを実行している間のサイクルを識別します。SYNCラインはOP CODEのフェッチの 約1の間に高レベルになり、そのサイクルの間、高レベルを保持します。SYNC 信号が高レベルになるうとする オクロックの間に、RD Y信号が落ちると、プロセッサーはその状態でストップし、RD Y信号が高レベルになるまで、その状態を保ちます。したがって、SYNC 信号はRD Y を制御して、一命令の実行に使用されます。

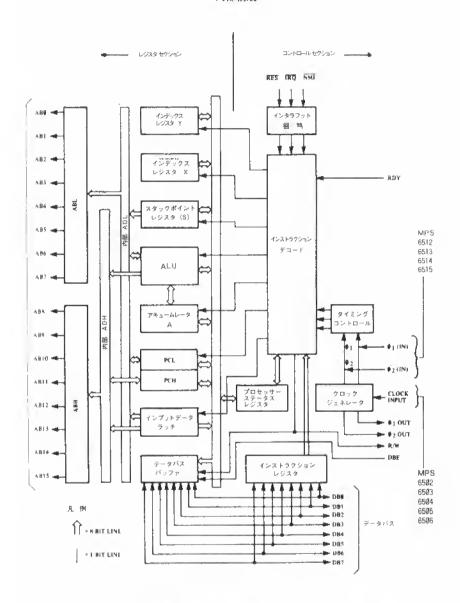
#### リセット (RES)

この入力は、マイクロプロセッサーをパワーダウン状態からのリセット、スタートに使われます。この信号が低レベルを保持している間は、マイクロプロセッサーからのまたはマイクロプロセッサーへの書きこみは禁止されます。立上がり入力があると、マイクロプロセッサーはすぐに、リセット・シークエンスを開始します。

システム初期化の6クロック・サイクルの後、割込マスクフラグがセットされ、メモリーのFFFC番地およびFFFD番地から、プログラム・カウンタをロードします。これが、プログラム制御の開始番地になります。パワーアップ・ルーチンでVccが4.75ボルトになった後、リセット信号は、少なくとも2クロック・サイクルの間、低レベルを保持されなくではなりません。この間にR/Wおよび(SYNC)信号が有効になります。

リセット信号が、その2クロック・サイクルの後、高レベルである間、マイクロ・プロセッサーは、上に述べた正常なリセット動作を続行します。

#### 内部構成



注意) 1,MPS 6512,13,14,15にはクロック・ジュネレータはありません。 2.アドレシング範囲とコントロール・オプションはMPS 65回側品により異なります。

命令セット-OP CODE、実行時間、サイズ

|                            |           | _             | _       | _    | _           |              |              |          |               |               | _              |          |               | _              |     | -        |          | _       | -      |        | -         |       |       |          |         |           |          |       |                |       | _                                |
|----------------------------|-----------|---------------|---------|------|-------------|--------------|--------------|----------|---------------|---------------|----------------|----------|---------------|----------------|-----|----------|----------|---------|--------|--------|-----------|-------|-------|----------|---------|-----------|----------|-------|----------------|-------|----------------------------------|
|                            | >         | 7             | 1       |      | 1           |              |              | Z,       |               | 1             |                |          |               | 1              |     | ۱.       |          | es-     |        | 1      |           |       |       | 1        |         | ٠.        |          | 1     | 1              |       |                                  |
| 4-10年の日本本の日<br>- 10年の日本本の日 | ۵         |               |         |      | •           |              |              |          | 1             | 1             |                |          | 1             | 1              |     | 6        | F        |         | 1      |        | -         |       | 1     |          | 1       |           | •        | 1     |                | 3     |                                  |
| 8                          | -         |               |         |      |             |              | ١.           |          |               |               |                |          | *             |                |     |          | 42       |         | *      |        |           |       |       |          | ,       |           |          |       | *              |       |                                  |
| +                          | U         | 3             | 4       | 7    | 1           |              |              | +        |               | 1             | -1             | 1        |               |                | 152 | .        |          | 1       | 5.     | 1      | 7         |       |       | 1        |         |           |          | 1     | 1              |       | 1                                |
| ñ :                        | ы         | 7             | 7       | 3    | 1           |              | ١.           | 7        |               |               |                |          |               | 1              |     | -        |          |         | 5      | 3      | 7         | 7     | 7     | 7        | 3       | 2         | 5        | 7     | 1              | 1     | 7                                |
|                            | z         | 2             | Y       | 7    |             | 1            | ١.           | ź        |               |               | . ]            |          |               | ×              |     | ٠.       |          |         | 3      | 1      | 2         | Σ     | 7     | 5        | 3       | 5         | 7        | 7     |                |       | 3                                |
| >-                         | 4         |               |         |      |             |              |              |          |               |               |                |          |               |                |     |          |          |         |        |        |           |       |       |          |         |           |          |       |                |       |                                  |
| かい シャ                      | z         |               |         |      | _           |              |              |          |               |               |                |          |               |                |     | 1        | _        |         |        |        |           |       |       |          |         | _t        |          |       |                |       |                                  |
| \$                         | ð         |               |         |      |             |              |              |          |               |               |                |          |               | -              |     |          |          | -       |        |        | - 1       | -     |       |          |         | _         |          |       |                |       |                                  |
| 4                          | -         |               |         |      | _           |              |              | -        |               | _             | $\neg$         |          | -             |                |     | $\dashv$ |          |         |        |        |           |       |       |          |         | $\neg$    | _        |       | ~              |       |                                  |
| 4094603                    | ÷e<br>Ζ   |               |         |      |             |              | -            |          |               |               |                |          | _             |                |     | $\neg$   |          | -       |        |        |           |       | _     |          | _       | $\dashv$  |          |       | 100            |       | _                                |
| 67.9                       |           | -             | _       | _    |             |              |              | _        |               | _             |                | -        |               | _              | _   | - †      | -        | _       | _      |        | $\exists$ | -     |       |          | _       | -         | _        | _     | ¥              | _     |                                  |
| Н                          | OF N # CF | $\vdash$      | _       | _    | ~           | ~            | ~            | _        | 04            | Pi            | ^1             |          | ~             | n              | _   | $\dashv$ |          | _       | _      | _      |           | _     | _     |          |         | $\dashv$  | _        |       | 40             | _     | _                                |
| 19517                      | -         | -             |         | -    |             | ~            | ~            | -        | N.            | ^-            | 24             | _        | ne .          | n <sub>4</sub> |     | $\dashv$ |          | _       | _      |        | -         | _     | _     |          | _       | -         | _        | _     | _              |       |                                  |
| 100                        | 4         | -             | _       | _    | 3           |              | h            |          | 9             | 8             | 2              | -        | 28            | 9              | _   | $\dashv$ |          |         | _      |        |           |       | _     |          |         | $\dashv$  |          |       |                | _     | _                                |
| 30                         |           | -             | _       | -    | 5           | 2            | 2            | _        | -             | <u>e</u>      | -              | -        | 7             | 6              | _   |          | _        |         | _      | _      | -         | -     |       |          | _       | -         | _        |       | _              | _     |                                  |
| $\frac{q}{-p}$             | m.<br>Z   | ~             | -       | _    | _           |              |              |          |               |               | -              |          |               | _              |     | $\dashv$ |          |         | _      |        | -         | -     | -     |          | ,-,m    | $\dashv$  | _        |       | _              |       | liles                            |
| 377                        |           | 76            | 皇       | _    | _           |              | -            |          |               | _             | -              |          |               |                |     | -        | _        | _       | 4      |        | _         | _     |       | _        | 7       | 4         |          | -     | _              | _     | Ţ.                               |
| ii.                        | ð         |               |         | _    | _           |              | $\vdash$     | _        |               |               | -              | _        |               |                |     | 4        |          | _       | 8      |        |           |       | _     |          | \$      | 4         | _        | _     |                | _     | £.                               |
| 14.6                       |           | -             | -       | _    |             |              | -            | _        |               |               |                |          |               | _              | _   | 4        |          |         | -      |        |           | E     | _     |          | ₩.      | ~         | _        |       |                |       | Mr.                              |
| 7729a+18 7759a-19          | z         | 4             | 4       | Ps.  |             | _            | _            |          |               |               |                | -        |               | _              |     | _        |          | _       | w      |        |           | Ь     |       |          |         | h         | _        |       |                |       | 7                                |
| Ä                          | ð         | 6             | Ģ       | ¥    | _           |              |              |          |               | _             |                |          |               |                |     | 4        | _        | -       | 8      |        | _         | 8     |       | _        | g       | Ξ         | _        |       | _              |       | 8                                |
| 15                         | =         | ~             | ~       | ~    |             |              |              |          |               |               |                | _        | _             |                |     |          |          |         | ~      |        |           | Pil   |       |          | N       | ~         |          |       |                |       | 140                              |
| 17-14                      | 2         | 4             | 4       | 0    |             |              |              |          |               |               |                |          |               |                |     |          |          |         | Ŧ      |        |           | 9     |       |          | ч       | ۵         |          |       |                |       | A116 2 81 5 2 85 4 2 80 4 3 99 4 |
|                            | ð         | 75            | 10      | £    |             |              |              |          |               |               |                |          |               | _              |     | _]       |          |         | å      |        |           | å     |       |          | 55      | £         |          | ,     |                |       | š                                |
| (4597,924)                 | 100       | r×            | Pa      |      |             |              |              |          |               |               |                |          |               |                |     |          |          |         | ~      |        |           |       |       |          | ~       |           |          |       |                |       | m                                |
| 0.4                        | Ż         | 10            | o,      |      |             |              |              |          |               |               |                |          |               |                |     |          |          |         | Vn.    |        |           |       |       |          | w       |           |          |       |                |       | ų.                               |
| 5                          | ð         | -             | Ξ,      |      |             |              |              |          |               |               |                |          |               |                |     |          |          |         | ō      |        |           |       |       |          | 5       |           |          |       |                |       | Ξ                                |
| i K                        | **        |               | 7       |      |             |              |              |          |               |               |                |          |               |                |     |          |          |         | N      |        |           |       |       |          | a       | $\neg$    |          |       |                |       | Pie                              |
| (47.2% 23.E)               | z         | ¢             | .0      |      |             |              | 1            |          | _             |               |                |          |               |                |     |          |          |         | 40     |        |           |       |       |          | .0      | $\exists$ |          |       |                |       | ф.                               |
| 3                          | ð         | -0            | -       |      |             |              |              |          |               |               |                |          |               |                |     |          |          |         | ā      |        |           |       |       |          | =       |           |          |       |                |       | 7                                |
|                            | ÷         | Ť             | Ť       | _    | _           |              |              |          | _             |               |                | -        |               | _              | -   | -        | -        | -       | Ť      | -      |           |       | _     |          |         |           | -        | -     |                |       |                                  |
| 422344                     |           | $\vdash$      | _       |      | _           |              |              |          |               |               |                | F        |               |                | Pi. | N        | n        | 2       |        |        | $\exists$ | -     |       | _        |         | 7         | -+       | P.    | _              |       | _                                |
| X.                         | *         | -             |         | _    | _           | _            | <del> </del> | _        |               | _             |                | 86       | -             |                | 90  | 8        | 3        | 2       | _      |        | -         | _     | J     | 2        | -       | $\dashv$  | 3        | 5     |                |       |                                  |
| 0.                         | N OF      |               | _       | _    | _           | _            | $\vdash$     | _        | _             | _             | $\exists$      | _        | _             |                |     | -        | _        | -       | _      | _      |           |       | Ų     |          | _       | $\dashv$  | _        | _     | _              | _     |                                  |
| 74,446.9                   | Z         | -             |         | ~    |             |              | 1            | -        |               | _             |                |          | -             |                | _   | $\dashv$ |          |         | _      | _      |           | -     | _     | _        | -       | -         | -        | -     | -              | _     | -                                |
| 44                         | m<br>he   |               |         | *    |             |              | -            |          |               | _             |                | -        |               | -              | _   | -        | -        | -       |        |        | $\dashv$  | -     |       |          |         | -         | _        |       |                | _     | -                                |
|                            | 9         | -             | -,      | - CA |             |              | -            | 7        |               | _             | -              | $\vdash$ | -             |                | _   | -        | -        | -       | Phy.   | ~      | -,        | ~     | _     | _        | Pq.     | 100       |          | _     | -              |       |                                  |
| 5-200                      | 40 # N    | -             | _       | 1/2  | _           | _            | -            | _        |               |               |                |          | _             | _              | _   | $\dashv$ |          | _       | _      |        | ~         | ~     | _     | _        |         |           | -        | _     |                | _     | _                                |
| 8                          |           | + 59          |         | _    | _           | _            |              | 74       | _             |               |                | -        | _             |                | _   | -        |          | -       | S      | _      | 5         |       | _     |          | 10      |           | -        | _     | _              | _     | 1 A5 J 2                         |
| H                          | ð         | ·9            | 3 25    | - 6  | -           | _            | $\vdash$     | 7        | _             | _             | -              | -        | _             | _              | _   | $\dashv$ | $\vdash$ | _       |        | Ξ<br>- |           | 3 (6  | _     | _        | 3 45    | 92        |          | _     | ,              | _     | *                                |
| +-E                        | u<br>Z    | +             | 7       | -    | _           | _            | -            |          |               | _             | -              | -        |               | -              |     | $\dashv$ | -        | _       | m<br>w |        | ~<br>+    | -     |       |          | m<br>=  | ~         |          |       |                | 9     | -                                |
| 77,771-4                   | Z         | 7             |         |      | _           | -            | -            |          | _             | _             | -              | _        | _             |                |     | $\dashv$ | _        | _       |        | 4      |           | -9    | _     |          | ~       |           | _        | _     | 11             |       | AV 2 2 AD 4                      |
|                            | ð         | 2             | 2 2 20  | ×    | _           |              | -            | ×        |               | _             |                | -        |               |                |     | $\dashv$ | -        |         | ŝ      | ¥      | QC.       | Ų.    |       |          | Ş       | #         |          | _     | ¥              | ₽,    | 7                                |
| イミ ティエイト                   | *         | 7             | ~       | _    | _           |              | -            | _        | _             | _             | _              | -        |               | _              |     | -        | _        |         | ~      | 7      | 7         | -     |       |          | ^4      |           | _        | _     | _              |       | N                                |
| 16                         | Z         | 100           |         |      |             |              | -            | _        |               |               | _              | _        | _             |                |     | _        | _        | _       | ~      | N      | ~         |       |       |          | a       | _         | _        | _     |                | -     | 14                               |
| 7                          | ð         | (4) is 69 2 2 | £,      |      | _           |              |              |          | _             |               |                | -        | _             |                |     | _        | _        |         | S      | 2      | ð         | -     | _     |          | Ç.      | _         |          | _     | _              |       | of the                           |
|                            |           | 2             | 1       |      | 9           | 9            | 5            |          | 5             | 9             | 9              |          | 5             | 9              |     |          |          |         | 1      |        |           |       |       |          | Ξ       |           |          |       |                |       | =                                |
|                            | ,         | 1 2           |         |      | -           | 7            | 5            |          | -             | 7             | 7              |          | ů,            | ī              |     |          |          |         |        |        |           |       |       |          |         | İ         |          |       | DOT MEN OJ AND |       |                                  |
|                            | A         | 4             | 4       | 1    | BRANCHENCES | MANCH ON C+1 | MANCH CALL   |          | BRANCH ON N#1 | BRANCH ON Z=0 | SEANCH ON NICE |          | BRANCH ON VER | MANCH ON VAI   |     |          |          |         |        |        |           |       |       |          | <       |           |          |       | 3              |       |                                  |
|                            |           | I F           | i       |      | S.          | 0            | OH           | -        | 0             | OH            | CH             |          | OH            | Ö              |     |          |          |         |        |        |           | 3     | ,,,,  | >        | 1 3 1 4 | 3 -1 - 1  | je<br>I  | -     | 0              | 13    |                                  |
|                            | Ė         | 0             | 75      |      | ¥           | X            | 13           | 3        | 3             | Ž             | y              |          | 3             | ¥              | 0-0 | 0+0      |          | 1       | 9-1    | 3      | N-4       | N-1-N | 1-1-1 | ¥-1-4    | 7       | -         | 1 -1 - 1 | 1+1+1 | -              | NA MA | 4                                |
| I                          | 11/11/13/ | 7 . 7         | 1       | - 1  |             |              |              |          |               |               |                |          |               |                | - 1 |          |          | 1       | T      | -      | - 1       | 1.1   | -     |          |         |           |          |       |                |       |                                  |
| ī                          | 144       | 4+44+6 - 4    | 1 3 . 4 | į    | 8           | 斯            | lif          | 4        | -             | -             | -              |          | 8             | M              |     |          | 100      | 01      | 4      | ef     | ř         | 3     | it    | 30       | 40      |           | per .    | gu    | 5              |       | 3                                |
| I                          |           |               | 0 4     | - 1  | 98          | 10           | ili<br>O     | <b>√</b> |               | 2             | 张              | -        | <b>B</b>      | 26             | 0   | •<br>0   | -        | -       | 4      | -      | j.        | 2     | 2     | je<br>je |         |           | 2        | -     | _              |       | 2                                |
| 1                          | 63        | Ų             | -       |      | _           | -            | •            |          | _             | _             | _              |          | _             | _              | _   | _        | 0 1 1    | ·       | -      |        |           | -     |       | _        | * 0     | 3         | -        | _     | 5              | * .   | ď                                |
| 1                          |           |               | Ċ       |      | Ų           | 10           | ď            |          | -             | -             |                |          | Ų             | uñ.            | _   | _        | -        | € 1 V € | ų.     | -      | je        | U     |       | _        | q       | U         | ~        | -     | ь              | w     | _                                |

|                                    | 43-7 (IA) 73          | 73.00a h | $\vdash$ | NO.     | ř. | 74.44 9 | $\vdash$ | 45/241 |    | 177.746 |           | W. S. T. T. T. T. | 14.8.6       | ×2->-  | - 15 | 100    | 77/11 - KK 77/10, KY | i k | 102   | 2            | 1135/7 | 5    | 3         | 447747       | $\vdash$ | ATT TO DA | -      |   | Estat video | 3   | Γ   |
|------------------------------------|-----------------------|----------|----------|---------|----|---------|----------|--------|----|---------|-----------|-------------------|--------------|--------|------|--------|----------------------|-----|-------|--------------|--------|------|-----------|--------------|----------|-----------|--------|---|-------------|-----|-----|
|                                    | 1                     | į.       | 1        | Н       | +  | ŀ       |          | 1      | 1  |         | +         | -                 | -            | ŀ      |      | -      | F                    | +   |       |              | -      | -    | 1         | 1            | +        |           | +      |   |             |     | 1   |
| 人をひしは後 めでは第一回は                     | Ž<br>Z                | z        | ð        | n.<br>Z |    | z<br>ð  |          | z<br>S | 흮  | z<br>ò  | -         | z<br>k            | O N          | ž<br>o | -    | z<br>ð |                      | ð   | Z     | *            | *      | Mac. | OF N F OF | z            | ð<br>*   | z         | z<br>m | 7 | _           | ۵   | >   |
| E                                  | A2 2 2 ME             |          | 92       | ~       | _  |         |          |        |    |         |           | _                 |              | _      |      |        | $\vdash$             | 36  | 4     | -            | H      | L    |           |              | Ź        | 4         | 3      | 7 |             |     | Γ   |
| ( D Y M-Y                          | 48 2 2 AC             | 4        | *        | 3       |    | _       |          |        |    |         | _         | _                 |              | 7      | m    | ¥      | -                    | _   |       |              |        |      |           |              |          | -         | 7      | 7 | ,           | 1   | ,   |
| 1 5 R a+· ·+C                      | *                     |          | \$       | 2 2     | 4  | 2       |          |        | _  |         |           | _                 |              | ·Ø     | 2    | ;      |                      | _   |       |              |        |      | 700001    |              |          |           | -      | 7 | 3           |     | ,   |
| V O P NO CHRATION                  |                       | _        |          |         |    | _       | ≤        | 24     | _  |         |           |                   |              |        |      |        |                      |     |       | _            |        |      |           |              | _        |           |        | 1 | ,           |     |     |
| O R A AVM-A                        | P9 2 2 80             | ~        | 58       | 3       |    | _       |          |        | 9  | ۵       | 7         | 2                 | 7            | 15:4   | PI   | 5      | m<br>=               | 3   | 7     | ,            |        |      |           |              |          |           | 7      | 7 |             | 1   | ,   |
| 7 H A A-W SI-5                     |                       | -        |          | H       |    | ┝       | \$       | -      | -  |         | $\vdash$  | $\vdash$          | F            | ┝      | ┞    |        | 1                    | ⊬   | L     | $^{\dagger}$ | ╁      | -    |           | ╁            | ╀        |           | +      | - |             |     | Т   |
| 7 + 1 + W + d d H d                |                       |          |          | _       |    | _       | 8        | m      | _  |         |           |                   |              |        |      |        |                      | _   |       |              |        |      |           |              |          |           | _      | - |             | 1   | ,   |
| P 1 A 5+1-5 M-A                    |                       |          |          |         |    | _       | 2        | ч      | _  |         |           |                   | _            |        |      | _      |                      |     |       |              | _      |      |           |              |          |           | ,      | 1 |             | -   | -   |
| P L P S+1-5 M-P                    |                       |          |          | _       |    | _       | 28       | +      |    |         |           |                   | _            |        |      |        |                      |     |       |              |        |      |           |              |          |           | _      |   | (回領)        | ~   |     |
| R O 1 = 0 = C+                     | 岩                     | . 9      | £        | 2 5     | ×  | 74      |          | _      |    |         |           |                   | _            | -9     | 74   | ×      | 7                    |     |       |              | _      |      |           |              |          |           | 7      | 7 | 7           | 1   | -   |
| # O k -C                           | 33                    | 9.       | 38       | 5 2     | 3  | 7       |          | -      |    |         |           | $\vdash$          | L            | 9 92   | ~    | ×      | - n                  | -   | L     | t            | ╁      | ╀    | İ         | <del> </del> | L        |           | 7      |   | 7           |     | Ţ   |
| R T I RTES INT                     | -,                    |          |          |         |    | _       | 3        | .0     | _  |         | _         |                   |              |        |      |        |                      | _   |       | _            |        |      |           |              |          |           | _      |   | 画           | _   | _   |
| R T S RTRN SUB                     | -                     |          |          |         |    |         | 2        | ·c     | _  |         | _         |                   |              |        |      |        |                      |     |       |              |        |      |           |              |          |           | _      |   |             |     | -   |
| 5 B C A-M-C+A (1)                  | F9 2 2 FD             | 7        | 2        | 79      |    | _       |          |        | Ξ  | ¢       | 01        | Ę I               | 2            | 4      | ~    | £      | 4                    | 5   | -     | Page 1       |        |      |           |              |          |           | 7      | 3 | 27          |     | ,   |
| 3-1                                |                       |          |          |         |    | _       | 3        | ž4     |    |         |           | _                 |              |        |      |        |                      |     | _     | ,            |        |      |           |              |          |           | _      |   | j           |     |     |
| _                                  |                       |          |          |         | _  |         | 9        |        | _  |         | _         |                   |              |        |      |        |                      | _   |       | _            |        |      |           |              |          |           | -      |   | -           |     |     |
| †                                  |                       |          | 1        | +       | 1  | +       | 2        | , ,    | +  | 1       | $\dagger$ | +                 | 1            | $^{+}$ | 4    | 1      | +                    | +   | Ţ     | †            | +      | 1    | 1         | +            | 4        | 1         | +      | 1 |             | -[  | .   |
| -                                  |                       |          |          |         |    |         | Ř        | ru.    | _  |         | _         |                   |              |        |      |        |                      | _   |       | _            |        | _    |           |              | .,       | -         | -      | , | -           | 1   | ,   |
| ¥14                                | G.                    |          | B 3      | 3 2     |    | -       |          |        | æ  | 4       | 7         | 9 5               | М            | 86     | 2    | g      | <b>M</b>             | \$  | 44    | ANT,         |        |      |           |              |          |           | 1      | 1 |             | ٠   | 1   |
| 2 1 x x 1 5                        | *                     | m<br>w   | 8        | 7       |    |         |          | -      | _  |         | -         |                   |              |        |      |        | _                    | _   |       |              |        |      |           |              | ₹        | 7         | ~      | 1 |             | 1   | ,   |
| N+V V T V                          | ×                     | 40       | 3        | 7       | _  |         |          |        | _  |         | -         |                   |              | 94     | ~    |        |                      |     |       |              |        |      |           |              | _        |           | '      |   |             | 1   | -   |
| 丁 4 耳 4一次                          |                       |          |          | _       |    |         | ş        | 7      | _  |         |           |                   | _            | _      |      |        |                      |     |       |              |        |      |           | -            |          |           | 7      | 7 | ,           |     | _   |
| 1 × 4 × 1                          | -                     | -        |          | _       |    |         | 2        | ~      | -  |         | -         | -                 |              | -      |      |        | <del> </del> −       | -   |       | t            | ╀      | L    | İ         | H            | L        | İ         | 1,     | 1 | ľ           |     | T   |
| T 5 X 5+X                          |                       | _        | _        |         |    | -       | Æ        | Ďi.    |    |         | _         |                   | _            |        |      | _      |                      |     |       | _            |        |      |           | -            | _        |           | 7      | 7 | i           |     | _   |
| 1 1 4 1 1 1                        |                       |          |          |         |    | _       | š        | ~      | _  |         | _         |                   |              |        |      | _      |                      |     |       |              |        |      |           |              |          |           | 7      | 7 |             | - 1 | - 1 |
| * * * × ×                          |                       | _        |          |         |    |         | ¥        | 2      | _  |         |           |                   |              |        |      |        |                      | _   |       |              |        |      |           |              |          |           |        |   |             |     | _   |
| 1 7 4 7 4 1                        |                       |          |          | -       |    |         | 9        | _      | _  |         | _         |                   |              |        |      |        |                      |     |       | -            |        |      |           |              |          |           | 7      | 3 | ,           |     |     |
| 御外                                 |                       |          |          |         |    |         | ^        | ,      | 1  | î       | ペログ 正子 シア | ,                 |              |        |      |        | 1                    | "   | 20 mm | ١,           | -      |      | 1         | 1            |          |           | -      | 1 | 3           |     | ]   |
| A STATE OF THE PARTY OF THE PARTY. | A. All Child had been | - An Oil | 4        | †       |    |         | -        | ,      | 1  |         | \$        | <                 |              |        |      |        | (                    |     | Ħ     | RC .         |        |      |           |              |          |           | 7      |   | サイクル数       | E S |     |
| ー、スーンスワンタンでがかった種のはRVA一品味られます。      | 178Mは日本               |          | 17       | a ·     |    |         |          | >      | +  | Į.      | インデックスと   | >-                |              |        |      |        | )                    |     | 論理和   | 1            |        |      |           |              |          |           | #      | 4 | バイト製        | -#K |     |
| 2. 四一ペーン内でのファンチの場のは Nに一名学されます      | の場合は N に              | 素と       | #<br>₩   | 6       |    |         | •        | ⋖      | P  | 77      | アキュムレータ   | P.                |              |        |      |        | >                    |     | 華     | 禁他的論理和       | 運和     |      |           |              |          |           |        |   |             |     |     |
| 3. キャリはボローと一致しません。                 | せん。                   |          |          |         |    |         |          | Σ      | K  | 1 A F   | 17        | 来効アドレスの内容         | 魯            |        |      |        | ۵                    |     | 一     | 室北岛印         |        |      |           |              |          |           |        |   |             |     |     |
| 4.10進モードでは、Zフラグは不正。                | は不正。                  |          |          |         |    |         |          | s<br>∑ |    | 000     | *         | 7                 | スタック・ボインタの内容 | 學      |      |        | I                    |     | 発     | 要化なし         |        |      |           |              |          |           |        |   |             |     |     |
| 簡果ゼロのチェックはアキュムレータでおこないます。          | コセレータで                | はいい      | #6       | 0       |    |         | '        | +      | 加斯 | Jani .  |           |                   |              |        |      |        | Σ                    |     | 4     | メモリーのビットフ    | 3      | 4    | 1         |              |          |           |        |   |             |     |     |
|                                    |                       |          |          |         |    |         | •        | ı      | 揮援 | Jani,   |           |                   |              |        |      |        | Σ                    |     | 14    | メモリーのビット6    | SE.    | 2    | ú         |              |          |           |        |   |             |     |     |

#### 命令セットーアルファベット順

- ADC メモリーとキャリーをアキュムレータに加算します。
- AND メモリーとアキュムレータの論理積をとります。
- ASL たへ1ビットシフト(メモリおよびアキュムレータ)します。
- BCC キャリーがOFFの場合、ブランチします。
- BCS キャリーがONの場合、ブランチします。
- BEQ 結果がゼロの場合、ブランチします。
- BIT アキュムレータの内容により、メモリーのビットテストをおこないます。
- BMI 結果がマイナスの場合、ブランチします。
- BNE 結果がゼロ以外の場合、ブランチします。
- BPL 結果がプラスの場合、ブランチします。
- BRK 強制プレイクします。
- BVC オーバーフローがOFFの場合、ブランチします。
- BVS オーバーフローがONの場合、ブランチします。
- CLC キャリーフラグをOFFにします。
- CDL 16進モードをクリアします。
- CLI 割込禁止フラグをクリアします。
- CLV オーバーフローフラグをクリアします。
- CMP メモリーとアキュムレータを比較します。
- **CPX** メモリーとインデックスレジスタXを比較します。
- **CPY** メモリーとインデックスレジスタYを比較します。
- DEC メモリーから1減算します。
- DEX インデックスレジスタ X から1 減算します。
- DEY インデックスレジスタYから1減算します。
- EOR メモリーとアキュムレータの「排地的論理和」をとります。
- INC メモリーに1加算します。
- INX インデックスレジスタXに1加算します。
- INY インデックスレジスタYに1加算します。
- JMP 新しいロケーションにジャンプします。
- JSR リターンアドレスをセーブして、新しいロケーションにジャンプします。
- LDA メモリーをアキュムレータにロードします。
- LDX メモリーをインデックスレジスタXにロードします。
- LDY メモリーをインデックスレジスタYにロードします。
- LSR 右へ1ビットシフト(メモリーおよびアキュムレータ)します。
- NOP 無操作
- ORA メモリーとアキュムレータの論測和をとります。

- PHA アキュムレータをスタックにブッシュします。
- PHP プロセッサー・ステータスをスタックにプッシュします。
- PLA スタックからテキュムレータにプルします。
- PLP スタックからプロセッサー・ステータスにプルします。
- ROL 左へ1ビット回転します(メモリーおよびアキュムレータ)。
- ROR 右へ1ビット回転します(メモリーおよびアキュムレータ)。
- RTI インタラプトからリターンします。
- RTS サブレーチンからリターンします。
- SBC アキュムレータからメモリーおよびボローを減算します。
- SEC キャリフラグをセットします。
- SED 10進モードにセットします。
- SEI 割込禁止フラグをセットします。
- STA アキュムレータをメモリーにストアします。
- STX インデックス X をメモリーにストアします。
- STY インデックス Y をメモリーにストアします。
- TAX アキュムレータをインデックスXに移します。
- TAY アキュムレータをインデックスYに移します。
- TSX スタックポインタをインデックスXに移します。
- TXA インデックス X をアキュムレータに移します。
- TXS インデックスXをスタックポインタに移します。
- TYA インデックスYをアキュムレータに移します。

# アドレシング・モード

アキュムレータ・アドレシング

1 バイト命令で、アキュムレータに対する操作です。

#### イミーディエイト・アドレシング

イミーディエイト・アドレシングでは、命令の2パイト目にオペランドがあり、メモリーのアドレスは必要ない。

#### アブソリュート・アドレシング

アブソリュート・アドレシングでは、命令の2パイト目が実効アドレスの下位8ビットで、3パイト目が上位8ビットになります。アブソリュート・アドレシングモードでは、アドレス可能メモリの65Kパイト全体のアクセスが可能です。

#### ゼロ・ページ・アドレシング

ゼロページ命令では、上位アドレスバイトを Ø として、命令の 2 バイト目の みをフェッチすることにより、コードと実行時間の節約をします。ゼロページを注意深く使えば、コードの効率が非常に増します。

#### インデックス・ゼロ・ページ・アドレシング(X、Yインデックス)

このアドレシングは、インデックス・レジスタに関連して使われ、「ゼロページのX」または「ゼロページのY」として参照されます。実効アドレスは、インデックスレジスタの内容に、第2パイトを加えることにより計算されます。「ゼロページ」アドレシングの形をとっているため、第2パイトの内容は、ゼロページを参照します。さらに、メモリの上位8ビットにはキャリは加えられず、ページ・パランダリーの交叉は起きません。

#### インデックス・アブソリュート・アドレシング(X、Yインデックス)

このアドレシングは、X、Yインデックスレジスタに関して使用され、 $\Gamma$ アブソリュートのX」または $\Gamma$ アブソリュートのY」として参照されます。実行アドレスは、X、Yの内容に、命令の第2、第3パイトの内容であるアドレスを加えることにより計算されます。このモードでは、インデックス・レジスタにはインデックスまたはカウント値を、命令には、ベース・アドレスを使用します。このタイプのインデックスでは、すべてのロケーションを参照でき、インデックスは、複数のフィールドを指すことができるので、コーディングと実行時間を減らすことができます。

#### インプライド・アドレシング

インプライド・アドレシング・モードでは、オペランドが持っているアドレスは、命令のオペレーションコード内にあります。

#### リラティブ・アドレシング

リラティブ・アドレシングは、ブランチ命令のみで使われ、条件ブランチに対して、デスティネーションを与えます。命令の第2パイトは、プログラム・カウンタが、次の命令をさした時、それの下位8ビットの内容に加えられたオフセットとして、オペランドになります。オフセットの範囲は、次の命令から、-128から+127パイトとなります。

#### インデックス・インダイレクト・アドレシング

インデックスは、インダイレクト・アドレシング (インダイレクト、X) では、 命令の第2パイトは、メインデックス・レジスタの内容に、キャリーを無視 して加算されます。その結果、ページ・ゼロのメモリーロケーションを指し、 その内容は、実効アドレスの下位8ビットになります。ページ・ゼロ内の次のメモリーロケーションの内容は、実効アドレスの上位8ビットになります。 実効アドレスの上下バイトを指定するメモリーロケーションは、いずれもページ・ゼロになくてはなりません。

#### インダイレクト・インデックス・アドレシング

インダイレクト・インデックス・アドレシング(インダイレクト、Y)では、命令の第2バイトは、ページ・ゼロのメモリーロケーションを指しています。そのメモリーロケーションの内容は、Yインデックス・レジスタの内容と加算して、実効アドレスの下位8ビットになります。キャリーは、ページ・ゼロ内の次のメモリーロケーションの内容と加算され、実効アドレスの上位8ビットになります。

#### アブソリュート・インダイレクト

命令の第2バイトの内容は、メモリーロケーションの下位8ビットになります。上位8ビットは、命令の第3バイトになります。そのメモリーロケーションの内容が実効アドレスの下位バイトであり、次のメモリーロケーションの内容が上位パイトになり、それが16ビットのプログラム・カウンターにロードされます。

#### 最大定格

| 項  | 目  | 話   | 号   | 定    | 格     | 単              | 位 |
|----|----|-----|-----|------|-------|----------------|---|
| 電源 | 電圧 | Vc  | 0   | -0.3 | ~+7.Ø | Vd             | c |
| 入力 | 電圧 | Vin |     | -0.3 | -+7.Q | V <sub>a</sub> | G |
| 動作 | 温度 | TA  |     | g -  | ~ 70  | Ĉ              |   |
| 保存 | 温度 | Ts  | r G | -5 - | -+15Ø | °C             |   |

# ●注意●

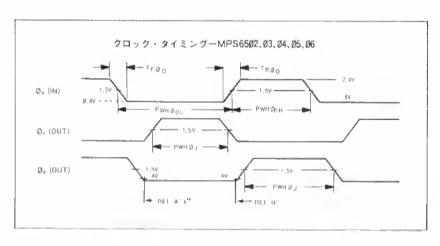
このチップは、静電気あるいは電界に対して保護されていますが、最大定格を超える電圧をかけないよう注意して下さい。

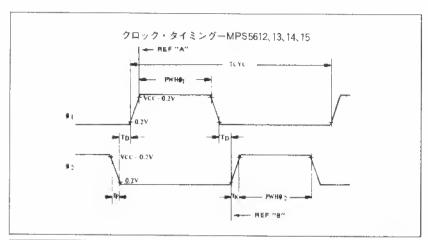
#### 電気特性 (Vcc=5.6V±5%、Vss= f, TA=25°C)

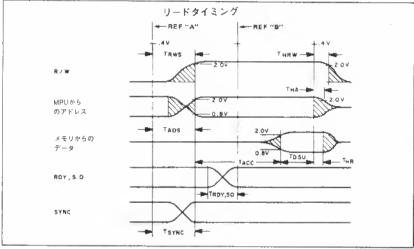
カ<sub>1</sub> 夕<sub>2</sub>はMPS6512、13、14、15に適用、カ<sub>3 (m)</sub>はMPS6502、03、04、05、96に適用

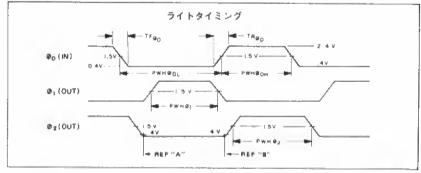
| 記号               | 項目               | 聂 小  | 標準  | 最大   | 単 位              | テスト条件   |
|------------------|------------------|--|-----|--|------------------|---|
| $V_{\rm IN}$     | /H^入 <b>か電</b> 圧 | V <sub>SS</sub> +2.4<br>V <sub>DC</sub> ←Ø.2 |     | V <sub>cc</sub><br>V <sub>cc</sub> +0.25     | Vds              | ロジック、タq(m)<br>タ1、タ2   |
| V <sub>II</sub>  | °L°入力電圧          | V <sub>55</sub> -0.3<br>V <sub>55</sub> -0.3 |     | V <sub>SS</sub> +Ø.4<br>V <sub>SS</sub> +Ø.2 | V <sub>dc</sub>  | ロジック、 $\mathscr{S}_{\mathbb{R}(m)}$ $\mathscr{S}_1$ 、 $\mathscr{S}_2$   |
| $V_{\rm HT}$     | *H*ヌレシュオールド電圧    | V <sub>SS</sub> +2.0                         |     |  | V <sub>dc</sub>  | RES, NMI, RDY, IRQ, Data, S.O.  |
| $V_{i\perp T}$   | "L"ヌレシュボールド電圧    |  |     | Vss+0.8                                      | V <sub>de</sub>  | RES, NMI, RDY, IRQ, Data, S.O.  |
| I <sub>I M</sub> | 入カリーク電流          |  |     | 2.5<br>190<br>197,9                          | μΑ<br>μΑ<br>μΑ   | $\{V_{\text{NN}} = \emptyset \text{ から5.25V}, V_{\text{CC}} = \emptyset\}$<br>ロジック (RDY, S.O以外)<br>$\mathring{p}_1$ 、 $\mathring{p}_2$<br>$\mathring{g}_{\text{obs}}$ |
| ITSI             | オフ状態スリーステート人力電流  |  | _   | 10   | μΑ               | (V <sub>in</sub> =∅.4から2.4V、V <sub>00</sub> =5.25V)<br>データライン   |
| V <sub>GH</sub>  | `H"出力電圧          | V <sub>55</sub> +2.4                         |     |  | V <sub>e</sub> , | (I <sub>LOAD</sub> = -100 #Ado, V <sub>CD</sub> = 4.75V)<br>SYNC, Data, A <sub>g</sub> = A <sub>15</sub> , R/W  |
| V <sub>0</sub> L | *L*出力電圧          |  |     | V <sub>5.5</sub> +0.4                        | V <sub>dc</sub>  | $(I_{LOAD}=1.6\pi Adv, V_{cc}=4.75V)$<br>SYNG, Data, $A_0-A_{rb}$ , R/W   |
| Po               | 消費電力             |  | .25 | .70  | W                |   |
| C                | 容量               |  |     |  | pF               | (V <sub>IN</sub> =Ø , T <sub>A</sub> =25℃ , f=1 MHz)  |
| Ġ <sub>nN</sub>  |                  |  |     | 19<br>15                                     |                  | ロジック<br>データ   |
| Cour             |                  |  |     | 12   |                  | Ag-As, R/W, SYNC  |
| C g a (in)       |                  |  |     | 50   |                  | ≶g (m)  |
| C yı             |                  |  | 30  | 5Ø   |                  | p <sub>1</sub>  |
| G #2             |                  |  | 5₽  | 8/7  |                  | <b>*</b> 2  |

注意 IROとNMIは、3Kのフルアップ抵抗が必要です。









注:REFはクロックのレファレンスポイントの意味です。

#### IMHzタイミング

#### クロック タイミングーMPS65!2, I3, I4, I5

| 記号               | 特性                                    |          | 最小         | 標準 | 最大 | 単位   |
|------------------|---------------------------------------|----------|------------|----|----|------|
| T <sub>CYC</sub> | サイクルタイム                               |          | 1000       |    |    | nsec |
| PWH ø1<br>PWH ø2 | クロックパルス幅<br>(Vori),2Vで測定)             | φ1<br>φ2 | 430<br>470 |    |    | nsec |
| T <sub>F</sub>   | 立下がり鮮間(g.2V~V <sub>cc</sub> -g.2Vで測定) |          |            |    | 25 | nsec |
| T <sub>D</sub>   | クロックの遅延時間(J.2Vで測定)                    |          | 0          |    |    | nsec |

## クロックタイミングーMPS6562,63,64,65,66

| 記号                              | 特性                               | <b>通</b> /js | 標準 | 最大                    | 単位  |
|---------------------------------|----------------------------------|--------------|----|-----------------------|-----|
| T <sub>CYC</sub>                | サイクルタイム                          | 1000         |    |                       | ns  |
| PWH <sub>p</sub>                | <sub>∮0:  M </sub> パルス幅(1.5Vで測定) | 460          |    | 520                   | ns  |
| ΤΚφο, ΤΕφο                      | ≠n(m)立上がり立下がり時間                  |              |    | 10                    | ns. |
| τ <sub>D</sub>                  | クロック運延時間(1,5Vで湯定)                | 5            |    |                       | ns  |
| PWHø <sub>1</sub>               | <b>≠</b> i(OUT)パルス幅(1.5∀で測定)     | PWHøoL-20    |    | PWHφ <sub>til</sub>   | ns  |
| PWHø <sub>2</sub>               | ≠2(00T)パルス幅(1.5Vで満走)             | PWH¢oH−40    |    | PWHφ <sub>oH</sub> 10 | ns  |
| T <sub>R</sub> , T <sub>F</sub> |                                  |              |    | 25                    | ns  |

#### リード/ライトタイミング

| 記号                | 特性                       | 表小  | 標準  | 最大  | 単位   |
|-------------------|--------------------------|-----|-----|-----|------|
| T <sub>RW5</sub>  | MPS65個からのリード/ライトセットアップ時間 |     | 100 | 300 | ns   |
| T <sub>ADS</sub>  | MPS6500からのアドレスセットアップ時間   |     | 100 | 300 | ľ)5  |
| T <sub>ACC</sub>  | メモリー・リードアクセス時間           |     |     | 575 | rii5 |
| T <sub>DSU</sub>  | データ安定時間                  | 100 |     |     | ns   |
| THR               | <b>陜込時、データホールド時間</b>     | 10  |     |     | ns   |
| THW               | 書込時、データホールド時間            | 30  | 60  |     | ns   |
| T <sub>MDS</sub>  | MPS6590からのデータセットアップ時間    |     | 150 | 200 | PIS. |
| TRDY              | RDY,S.O.セットアップ時間         | 100 |     |     | ris  |
| T <sub>SYNC</sub> | MPS65ØbからのSYNGセットアップ時間   |     |     | 350 | ris  |
| THA               | アドレスホールト時間               | 30  | 60  |     | FFS  |
| T <sub>HRW</sub>  | R/Wホールド時間                | 30  | 60  |     | ns   |

#### 2MHzタイミング

# クロックタイミングーMPS6512、13、14、15、16

| 記号               | 特性                                      |          | 最小         | 標準       | 最大 | 単位   |
|------------------|---|----------|------------|----------|----|------|
| T <sub>CYC</sub> | サイクルタイム                                 |          | 500        |          |    | nsec |
| PWH ø2           | クロックバルス幅<br>(V <sub>00</sub> = 0.2Yで測定) | φ1<br>φ2 | 215<br>235 |          |    | nsec |
| T <sub>F</sub>   | 立下がり時間(Ø.2V~V <sub>co</sub> 一Ø.2Vで測定)   |          |            | <b>†</b> | 12 | nsec |
| T <sub>D</sub>   | クロック遅延時間(M.2Vで測定)                       |          | 0          |          |    | nsec |

# クロックタイミングーMPS6562,63,84,85,66

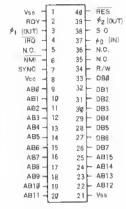
| 記号  | 特性   | 長小                     | 標準 | 最大                      | 単位  |
|---|--|------------------------|----|-------------------------|-----|
| T <sub>CYC</sub>  | サイクルタイム  | 500                    | -  |                         | ns  |
| PWH <sub>o</sub>  | ச <sub>ராவு</sub> パルス幅(1,5Vで測定)                          | 240                    |    | 260                     | ns  |
| $TR\phi_{\scriptscriptstyle O}$ , $TF\phi_{\scriptscriptstyle O}$ | ≠ <sub>O(Ne)</sub> 立上がり立下がり時間                            |                        |    | 10                      | ns  |
| TD  | クロック遅延時間(1.5Vで測定)  | 5                      |    |                         | ns  |
| PWHø <sub>1</sub>   | ! <sub>1(001)</sub> パルス橋(1.5Vで測定)                        | PWH <sub>ot</sub> -20  |    | PWH pol                 | ńs  |
| PWHø <sub>2</sub>   | ≠ <sub>210√11</sub> パルス幅(1.5Vで測定)                        | PWHφ <sub>oH</sub> −40 |    | PWHφ <sub>oH</sub> - 10 | 175 |
| T <sub>R</sub> , T <sub>F</sub>                                   | チ:cam &:com 立下がり時間<br>(*0.8Vから2.0で測定)<br>(ロード−3ØpF+1TTL) |                        |    | 25                      | nş  |

#### リード/ライトタイミング

| 紀号               | 特性                        | 最小 | 標準  | 最大  | 単位  |
|------------------|---------------------------|----|-----|-----|-----|
| TRWS             | MPS656Mからのリード/ライトセットアップ時間 |    | 100 | 150 | ns  |
| T <sub>ADS</sub> | MPS65船Aからのアドレスセットアップ時間    |    | 100 | 150 | ns  |
| T <sub>ACC</sub> | メモリーリードアクセス時間             |    |     | 300 | ns  |
| T <sub>DSU</sub> | データ安定時間                   | 50 |     |     | ńs  |
| THR              | 読込時データホールド時間              | 10 |     |     | ns. |
| THW              | 書込時データホールド時間              | 30 | 60  |     | ns  |
| T <sub>MD5</sub> | MPS65側Aからのデータセットアップ時間     |    | 75  | 100 | ns  |
| TREEY            | RDY, S.O. セットアップ時間        | 50 |     |     | ns  |
| TSYNC            | MPS6580AからのSYNCセットアップ時間   |    |     | 175 | ns  |
| THA              | アドレスホールド時間                | 30 | 60  |     | ns  |
| THRW             | R/Wホールド時間                 | 30 | 60  |     | П5  |

### ピン バッケージ

#### MPS6502



### 特徵

- ●65Kバイト アドレサブル
- IRO 割込
- NMI割込
- ●内蔵クロック TTLレベル単相入力 RCベースタイマ入力 クリスタルベースタイマ入力
- ●同期信号 (命令実行に用いられる)
- ●レディ信号 (†サイクル実行に用いられる)
- ●サポートチップに対し2相クロック出力

# MPS-656Ø ビデオ・インターフェイス・チップ(VIC)

# 概要

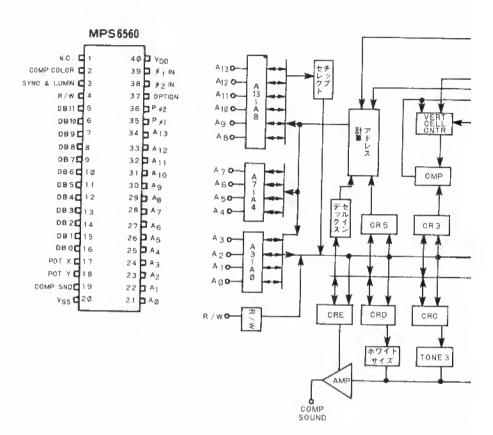
MPS656Øビデオ・インターフェイス・チップ(VIC)は、低コストのCRT端末、生医学的なモニタ、コントロール・システム・ディスプレイ、アーケード/ホームビデオ・ゲーム、ホーム・コンピューターなどのようなカラー・ビデオ・グラフィック応用製品のために設計されています。カラーでプログラマブルなキャラクターグラフィックを高分解能で生成する回路を提供します。また、VICは、ビデオ・ゲームに用いられるA/Dコンパータと音声効果を提供します。

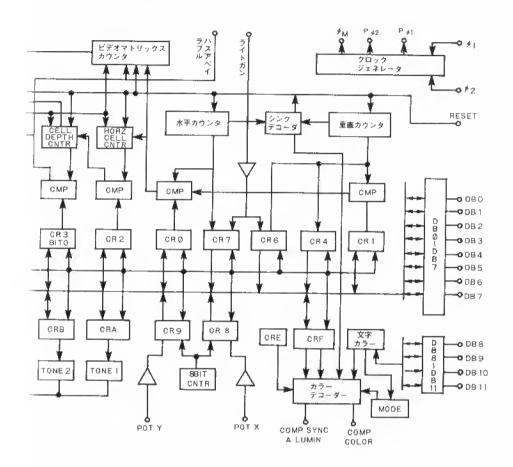
チップの音声システムには、3つの独立で、プログラマブルな音声発生器、ホワイト・ノイズ発生器、振幅変調器があります。したがって、スクリーンのリフレッシュの間のCPUの空時間をなくし、プログラマブルなスイッチにより、組合せ、非組合せの操作をすることができます。MPS656頃は、2種類のモードで、カラー操作します。

# 特長

- ○16Kバイトアドレス空間に完全拡張可能
- ○マスク・プログラマブル同期ジェネレーション
- ○オン・チップ・カラージェネレーション
- 6回のプログラマブルでムーバブルなバックグランド・ロケーション
- ○192×200のスクリーン・グリッド
- ○キャラクタ・サイド2種
- ○オン・チップ・音声システム
- ○オン・チップ・DMA、アドレス・ジェネレーション
- ○16アドレス可能コントロール・レジスタ
- ターゲット・ゲームにはライトガン / ペン使用可能

# ビン配置図





# 信号の説明

### アドレスバス (AØ-A13)

14ビットのアドレスパス(A Ø - A 13)は双方向性です。P  $\underline{A}$  = 1 の間、アドレス・ピンは人力モードであり、マイクロプロセッサーは、VIC の16のコントロール・レジスタのいずれもアクセス可能です。アドレスパスの上位ピン(A 8 から A 13)は、チップセレクトの役割をします。実際のチップ・セレクトは、A 13 = A 11 = A 1 Ø = A 9 = A 8 = Ø で A 12 = 1 の場合で、これは、VICのチップ・セレクト・アドレスが、16進の1969の場合になります。アドレスパスの下位 4 ビット(A Ø から A 3)は、入力アドレスによるコントロールレジスタ選択に使われます。

P 角 = 1・の間は、VICアドレスピンは、データ(キャラクタ・ポインタ、あるいは、キャラクタ・セル)がフェッチされている場合は、出力モードになります。 VICはフェッチされているメモリー・ロケーションのアドレスを出力します。VICのアドレスは、P 角のエッヂの立上がりから、50msで有効になり、P 角のエッジの立上がりまで続きます。

### リード/ライト (R/W)

この信号は入力のみで、VICとマイクロプロセッサーとのデータの流れを制御します。R/W信号が低レベルで、VICのチップセレクト条件が満足されている場合、マイクロプロセッサーは、選択されているVICコントロール・レジスタにデータの書込みが可能です。R/W信号が高レベルで、チップセレクト条件が満たされている場合、マイクロプロセッサーは、選択されているVICコントロール・レジスタからデータの読込みが可能です。

### データバス (DBØ-DBII)

12ビットデータバス(DBØ一DB11)は、2つの部分に分けられます。下位8ビット(DBØからDB7)はマイクロプロセッサーとディスプレイに必要なデータのインタフェイスに使われ、上位4ビットは色とモード情報に使われます。下位8ビット(DBØからDB7)は、さらに2つの分類に分けられます。すなわち、マイクロプロセッサー側のインタフェイスとビデオデータ側のインタフェイスの2種類です。 $P\mathfrak{A}=1$ の間、DB7からDBØはマイクロプロセッサーVICのデータの受渡しに使われます。 $P\mathfrak{A}=1$ の間は、DB7からDBØはディスプレイデータのフェッチに使われます。

### クロック

### マスタ・オシレータ・クロック入力(∮1、∮2)

656Øは14.31818MHz(NTSC)、2相クロックが必要です。クロック信号は、十5 Vで非重複でなくてはいけません。

### システムクロック (Ph. Ph)

これは、VICのタイミング・ジェネレータです。+5 V、非重複、1.02MHzで、6512マイクロプロセッサのドライブが可能です。

### メモリークロック (オプション、 øм)

これは単一2.Ø4MHzクロックで、アドレスバスが有効になった後、VICのメモリーに、ストローブが必要な場合、用いられます。37ピンに設置可能なオプションです。

### アナログ・ディジタルコンバータ(POTX、POTY)

これは人力ピンで、マイクロプロセッサーの判別できる8ビットの16進数に、 分圧値を変換します。単一RC時定数積分により実行されます。分圧計は、ボット・ピンに接続された外部コンデンサの測定に用います。

### コンポジット・サウンド (COMP SND)

このピンは、VICブロック図に示すように、6560の音の合成を出力します。 ハイ・インピーダンス出力(約1  $K\Omega$ )で、スピーカーをドライブするには、外部 に増幅器およびバッファを必要とします。

# コンポジット・シンクおよびルミナス (SYNC & LUMIN)

このピンはオープン出力で、標準テレビジョンに必要なビデオ同期と輝度に関する情報を与えます。

# コンポジット・カラー (COMP COLOR)

この信号は、標準テレビジョンが受けとる色の情報を与えます。コンポジット・カラー・ピンは、ハイ・インピーダンス出力バッファであり、3.579545MHzで、 突発信号とカラー逆変換および振幅情報を与えます。

### リセット

オプションの37ピン人力で、水平・垂直同期カウンタを外部信号と同期をとります。

### バス・アベイラブル

オプションの37ピン出力で、ビデオのメモリー・フェッチに対して、VICの状態を示します。VICがメモリーアクセスをおこなう 2 μ sec前に低レベルになり、 スクリーンがリフレッシュされるまで、低レベルを保ちます。

### ライト ガン/ペン

オプションの37ピン入力で、立下がりにより、コントロールレジスタ6、7にラッチされているスクリーン上にスキャンされている瞬間のドット位置を与えます。このピンは、「ターゲット・ショット」等のゲームやライトペンに用いられるフォトディテクターに使われます。

# 動作説明

プログラマブル・カラー文字をつくるために、VICは、3つに分かれたエリア、すなわち文字ポインタ、ディスプレイ文字、カラーポインタから、外部メモリのアクセスをおこないます。文字ポインタはRAMの1ブロック(ビデオ・マトリックスとよばれる506パイトが代表的)であり、そこに表示したい文字のポインタがあります。文字エリアは、8か16パイトのブロックからなっており(通常セルとよばれる)そこに表示しようとする実際のドットパターンがあります。文字セルはRAMでもROMでもかまわず、スクリーンに表示する方法によります。カラーポインタ・エリアはRAMの小さなブロックにあります(ふつう506個の4ビット群がカラー・マトリックスとよばれます)。4ビットカラーポインターは表示文字の色を決定し、2つのモードのうち1つを選択します。

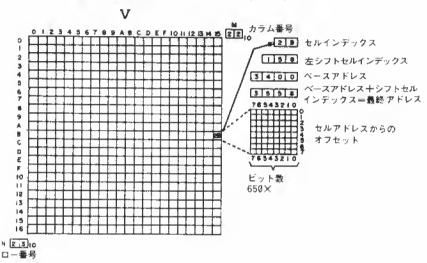
ビデオ・マトリックス、カラー・マトリックス、文字セルを結びつけて、必要 なデータをスクリーンに表示するのは、外部のマイクロプロセッサーによります。

VICの動作についてより完全に理解するために図1を示します。典型的なビデオ・マトリックスで、水平方向に22文字、垂直方向に23文字表示可能で、スクリーンの水平方向176ドット、垂直方向184ドットに対して506文字を表示します。文字表示位置は、対応するポインタまたはインデックスをもち、特定の位置に文字を表示します。

たとえば、(B、15)には 2 Bというインデックスがあります。つまり 2 B番目の文字が、その位置に表示されるのです。VICは文字インデックス値 2 Bをフェッチし、表示する文字位置のアドレス計算をおこないます。計算方法は簡単で、 $8\times8$  の文字セルの場合、インデックスは 3 回左にシフトレ (8 倍)、VICコントロールレジスタ C R 5 にあるスタートアドレスを加算します。今の場合、文字セルの開始アドレスは 3 34990であり、文字インデックスの左シフトした値と加算され、文字位置として、メモリ上 3 3558が得られます。

ある特定文字の表示回数は無制限です。同じ文字インデックス(たとえば2B)を使用して、文字データは何度でも表示されます。簡単なソフトウェア・ドライバにより、十分なRAM(約4KバイトのセルRAM)があれば、ビット単位の表示システムとしてVICを使うことができます。

図1 典型的なビデオマトリックス(23×22)



### 補色/バックグラウンド・カラー

- Ø ブラック
- 1 ホワイト
- 2 レッド
- 3 シアン
- 4 マジェンタ
- 5 グリーン
- 6 ブルー
- 7 イエロー
- 8 オレンジ
- 9 ライトオレンジ
- A ピンク
- B ライトシアン
- C ライトマジェンタ
- D ライトグリーン
- E ライトブルー
- F ライトイエロー

### ボーダー/キャラクター・カラー

- Ø ブラック
  - 1 ホワイト
  - 2 レッド
  - 3 シアン
  - 4 マジェンタ
  - 5 グリーン
  - 6 ブルー
  - 7 410-

# レジスタ機能

656頃には8ビット構成のコントロール・レジスタが16間あり、マイクロプロセッサーがVICのすべてのオペレーティング・モードでのコントロールを可能にしています。コントロール・レジスタとその機能については次に記述します。図2はレジスタのロケーションと内容を示しています。

| 1000            | ORIC | SIN                         | 図 2                         | VIC                         | コントロ                         | コールレ                        | ・ジスタ                         |                  |                             |                     |
|-----------------|------|-----------------------------|-----------------------------|-----------------------------|------------------------------|-----------------------------|------------------------------|------------------|-----------------------------|---------------------|
|                 |      | 7                           | 6                           | 5                           | 4                            | 3                           | 2                            | 1                | 0                           | ビット番号               |
| $CR_0$          | 1000 | ı                           | Sxb                         | S <sub>3</sub> ,5           | S <sub>X</sub> 4             | 5 <sub>X</sub> 3            | S <sub>X</sub> <sup>2</sup>  | S <sub>X</sub> ¹ | SXO                         | スクリーンX座標<br>限点      |
| CR <sub>1</sub> | 1001 | S <sub>V</sub> 7            | Sy6                         | Sys                         | 5 <sub>V</sub> 4             | Sys                         | 5 <sub>Y</sub> 2             | S <sub>V</sub> 1 | S <sub>Y</sub> 0            | スケリーンY座標<br>原点      |
| CR <sub>2</sub> | 1002 | B <sub>V</sub> 9            | Mb                          | Ms                          | M <sub>4</sub>               | Мз                          | M <sub>2</sub>               | М1               | M <sub>0</sub>              | ビデオマトリックス<br>列番号    |
| CR <sub>3</sub> | 1003 | R <sub>O</sub>              | N <sub>5</sub>              | N <sub>4</sub>              | N <sub>3</sub>               | N <sub>2</sub>              | N <sub>1</sub>               | N <sub>0</sub>   | D                           | ビデオマトリックス<br>行番号    |
| CR <sub>4</sub> | 1004 | R <sub>B</sub>              | R <sub>2</sub>              | R <sub>b</sub>              | R <sub>5</sub>               | R <sub>4</sub>              | R <sub>3</sub>               | R <sub>2</sub>   | R <sub>1</sub>              | ラスター値               |
| CR <sub>5</sub> | 1005 | B <sub>V</sub> 13           | B <sub>V</sub> 12           | 8 <sub>V</sub> 11           | B <sub>V</sub> <sup>10</sup> | B <sub>C</sub> 13           | B <sub>C</sub> <sup>12</sup> | BcII             | B <sub>C</sub> 10           | ベースアドレス<br>コントロール   |
| $CR_6$          | 1006 | L <sub>H</sub> <sup>7</sup> | L <sub>H</sub> 6            | L <sub>H</sub> S            | L <sub>et</sub> 4            | L <sub>H</sub> 3            | L <sub>H</sub> <sup>2</sup>  | L <sub>H</sub> ¹ | L <sub>H</sub> 0            | ライトペン<br>水平         |
| CR <sub>7</sub> | 1007 | L <sub>V</sub> <sup>7</sup> | L <sub>V</sub> 6            | L <sub>V</sub> 5            | L <sub>V</sub> 4             | L <sub>V</sub> 3            | L <sub>V</sub> ?             | L <sub>V</sub> 1 | L√0                         | ライトペン<br>手直         |
| $CR_{\sigma}$   | 1008 | P <sub>1</sub> 7            | P <sub>X</sub> 6            | P <sub>X</sub> S            | P <sub>X</sub> 4             | P <sub>k</sub> <sup>3</sup> | P <sub>K</sub> ?             | P <sub>N</sub> 1 | p <sub>x</sub> o            | ボットX                |
| CR <sub>4</sub> | 1009 | $P_{Y}^{7}$                 | P <sub>V</sub> 6            | P <sub>Y</sub> <sup>5</sup> | Py4                          | P <sub>V</sub> 3            | P <sub>Y</sub> <sup>2</sup>  | P <sub>Y</sub> 1 | ₽ <sub>V</sub> ô            | ボットY                |
| $CR_A$          | 100A | S <sub>1</sub>              | F <sub>1</sub> 6            | F <sub>1</sub> <sup>5</sup> | F <sub>1</sub> 4             | F <sub>1</sub> J            | F <sub>1</sub> <sup>2</sup>  | F <sub>3</sub> 1 | F <sub>1</sub> 0            | F <sub>IN</sub> (1) |
| $CR_B$          | 1008 | 52                          | F26                         | F <sub>2</sub> 5            | F24                          | F <sub>2</sub> 1            | F <sub>2</sub> 2             | F <sub>2</sub> 1 | F20                         | F <sub>IN</sub> (2) |
| $CR_C$          | 100C | S <sub>3</sub>              | F36                         | F <sub>3</sub> 5            | F34                          | F <sub>3</sub> <sup>3</sup> | F <sub>3</sub> 2             | F <sub>3</sub> 1 | F <sub>3</sub> <sup>0</sup> | F <sub>PV</sub> (3) |
| $CR_D$          | 100D | 54                          | F <sub>4</sub> 6            | F <sub>4</sub> 5            | F <sub>4</sub> 4             | Fall                        | F <sub>4</sub> <sup>2</sup>  | F <sub>4</sub> 1 | F40                         | F <sub>[m</sub> (4) |
| $CR_{I}$        | 100€ | C <sub>A</sub> 3            | C <sub>A</sub> 2            | $\mathbb{C}_{\mathbb{A}^1}$ | C <sub>A</sub> O             | .A <sub>1</sub>             | A <sub>2</sub>               | A <sub>1</sub>   | A <sub>0</sub>              | 振幅                  |
| $CR_f$          | 100F | C <sup>B</sup> 3            | C <sub>B</sub> <sup>2</sup> | C <sub>B</sub> <sup>1</sup> | Cg <sup>0</sup>              | R                           | C42                          | C <sub>f</sub> 1 | C <sub>f</sub> <sup>0</sup> | カラー<br>コントロール       |

注) NU -未使用

**2** 2

### CRØ

ビットサー6はテレビ・スクリーンの左端から最初に現われる文字カラムの距離を決定し、スクリーンのビデオ・マトリックスの横方向のサイズに使用されます。ビット7はインターレース・スキャン・モード(1=1)を選択します。

### CR1

テレビ・スクリーンの上から最初に現われる文字ローの距離で、スクリーンの ビデオ・マトリックスの垂直方向のサイズに使用されます。

#### CR2

ビット $\emptyset$  -6 はビデオ・マトリックスのカラム数をセットします。ビット7は、CR5のビデオ・マトリックス・アドレスの一部です。

#### CR3

ビット1-6はビデオ・マトリックスのローの数をセットします。ビット $\emptyset$ は $8\times 8$ の文字マトリックス( $D=\emptyset$ )か $16\times 8$ の文字マトリックス(D=1)に使用されます。ビット7はCR4のラスター値の一部です。

#### CR4

テレビのラスター・ビームがスキャンする現在のライン・ナンバーを示します。

#### CR5

ビット Ø - 3 は文字のセル・スペースがはじまるアドレスを決めます (ビット A 13から A 1Øはアクチュアル・アドレスです)。ビット4 - 7および C R 2 のビット 7 はビデオ・マトリックスのスタート・アドレスを決めます (ビット A 13から A 9 はアクチュアル・アドレスです)。

#### CR6

ライトペンの水平方向のラッチ・ポジションです。

#### CR7

ライトペンの垂直方向のラッチ・ポジションです。

### CR8

POTXのデジタル値です。

#### CR9

POTYのデジタル値です。

#### CRA

ビット $\emptyset$ - $\delta$ は最初のオーディオ・オシレーターの周波数をセットします。 ビット7はオシレーターがオンで1、オフで $\emptyset$ です。

#### CRB

2番目のオーディオ・オシレーター用で、CRAと同様です。

#### CPC

3番目のオーディオ・オシレーター用で、CRAと同様です。

#### CRD

ノイズの周波数をセットし、CRAと同様です。

### CRE

ビット $\emptyset$ -3は、オーディオ・シグナルのボリュームをセットします (いかなる音を作る時も最小1つのサウンド・ジェネレーターはオンの状態にしておかねばなりません)。ビット4-7はマルチカラー・モードのオペレーションで使用される補色のコードを表わしています。

189

### CRF

ビット 4-7は16種のバックグラウンドのカラーのうち1つを選択し、ビデオ・マトリックス内のバックグラウンド・エリアにそのカラーをセットします。ビット  $\emptyset-2$ はビデオ・マトリックスの外側のボーダーエリアに 8色のうち1つを選択します。ビット 3はビデオ・マトリックスで共通のバックグラウンド・カラーのとき R=1で、反転するときは  $R=\emptyset$ です。全文字を同じバックグラウンド・カラーにしたり、個々異なるバックグラウンドにするのは、カラーRAM内のコードによって決めます(マルチカラー・モードのときは R ビットは関係なく、CRF機能もこのモードでは異なります。詳細はオペーレーティング・モードのセクションを参照して下さい)。

### カラー・オペレーティング・モード

VICには2つのカラー・オペレーション・モード、HI-RES (ハイ・レゾリューション) モードとマルチカラー・モードがあります。このオペレーティング・モードは、基本的にはどのようにしてキャラクター・セルに情報を与えてテレビ・スクリーン上の点を変化させるかを決めます。オペレーティング・モードは、ビデオ・マトリックスの個々のキャラクター・ロケーションで示されるカラー・ポインターのMSBで決定されます。もしキャラクターのカラー・ポインタのMSBがゼロの時は、そのキャラクターはHI-RESモードで表示されます。MSBが1になると、そのキャラクターはマルチカラー・モードで表示されます。

HI-RESモードではキャラクター・セルとスクリーン上の表示される点が1対1に対応しています。たとえばキャラクターの全ピットが1のとき表示される色もあれば、全ビットがゼロの時に表示される色もあります。文字のフォーグラウンド・カラーはキャラクター・カラー・ポインターの残り3ビットで指定され、バックグラウンド・カラーはCRFで指定されます。

マルチカラー・モードではキャラクター・セルの個々2ピットがスクリーン上の1点と対応し、その点の色は2ピット・コードで決まります。HI-RESモードは1つの文字では2色までしか表示できません。マルチカラー・モードは1つの文字で4色まで可能ですが、これはセル・データの2ピットがスクリーン上の1点と対応しているためで、HI-RESモードを水平に半分したにすぎません。これは8×8のキャラクター・セルのメモリー・マップをスクリーン上で8×4のキャラクターにしたことになります。ここで注意しなければならないのは、8×4のマルチカラー・キャラクターと8×8のHI-RESキャラクターの要求するメモリーの量は同じであり、単にスクリーン上のマップが異なるだけであるということです。

マルチカラー・モードでは2ビットが点を構成する4色のうちの1色を選択します。これらの2ビットで構成される4つのコードがVICに点のカラー情報を与えます。これにより、その点の色はバックグラウンド、ボーダー、補色、フォーグラウンドのどれかになります。

マルチカラー・モードで選択されるコード

●注意● 2ビットのコード自身はカラー・コードではなく、3または4ビットで構成される4つの異なったカラー・コードのポインターにすぎません。

### 例)

### 文字の定義

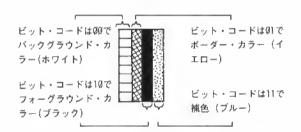
|     |   | レット      |     |
|-----|---|----------|-----|
|     |   | 76543210 | 16進 |
| バイト | ø | ØØØ11Ø11 | 1B  |
|     | 1 | 99911911 | 1B  |
|     | 2 | 99911911 | 1B  |
|     | 3 | 90011011 | 18  |
|     | 4 | ØØØ11Ø11 | 18  |
|     | 5 | ØØØ11Ø11 | 1B  |
|     | 6 | 00011011 | 1B  |
|     | 7 | 00011011 | 1B  |

ما بورتما

もしカラー・ポインターが指定する文字が Ø(ØØØØ)のときは、フォーグランド・カラーはブラック(Ø)でHI-RESモード(MSB=Ø)が選択されています。この文字は次のようにスクリーンに表示されます。



もしカラーの指定している文字が8 (1 $\emptyset$  $\emptyset$  $\emptyset$ ) のときは、フォーグラウンド・カラーはブラック ( $\emptyset$ ) でマルチカラー・モード (MSB=1) が選択されています。この文字は次のようにスクリーンに表示されます。



●注意● これは単なる例であり、多くのテレビ・セットでは接近した点の異なる色は鮮明には出ません。

文字の表示モードはキャラクター・カラー・ポインターで指定され、スクリーン上のキャラクター・ロケーションは個々独立したカラー・ポインターを持っているので、HI-RESとマルチカラー・キャラクターを自由にまぜることが可能です。これによりアルファニュメリックをHIS-RESモードのキャラクターで表示でき、マルチカラー・モードでは文字の幅を広くすることもできるなど、表示機能の柔軟性を大幅に増大しています。

# VICコントロールレジスタの使用例:

簡単にするために全部の文字がHI-RESモードであり、VICレジスタは次の値に位置するとします。

| レジスター | 内容(16進)  | 2        | 進    | 結   | 果     |
|-------|--|----------|------|---|-------|
| CRØ   | Ø3   | 9/990    | ØØ11 | スクリーンのビデス<br>端から 3×4ドット報<br>インターレースは選択                |       |
|       |  |          |      | =0),  |       |
| CRI   | 19   | ØØØ1 1Ø  | 91   | スクリーンの上から<br>スを16進の19×2ドッ<br>します。                     |       |
| CR2   | 96   | 1/001 01 | 10   | ビデオ・マトリック<br>16 (=22、ベースは1<br>(ビット7はCR5で              |       |
| CR3   | 2 E  | X/Ø10 11 | 1/0  | ビデオ・マトリック<br>(=23、ベースは10)<br>8×8キャラクターの<br>れます (D=0)。 |       |
| CR5   |  | スのスタート・  | アドレス | をアクセスためにセット<br>が16進の34何のになってい。                        |       |
| C R 5 | Ø D 14ビットのアドレスで構成されます。 CR®ビット CR®ビット CR®ビット CR®ビット 90 001 X XXXX XXXX 0 2 0 0 2 れはビデオ・マト リックス用です。 キャラクスも同様に14 ビットで構成されます。 CR®ビット 32 10 11 01XX XXXX XXXX 3 4 0 0 | ØØØØ 11  | Ø1   | C R 2のビット7 は 1  | にセット。 |

| CRA | ØØ  | Ø/999 9999   | オシレーター1はオフ。                              |
|-----|-----|--------------|--|
| CRB | 9 A | 1/901 1919   | オシレーター2はオンで、相対周波数は1 A。                   |
| CRC | Ø Ø | 0/000 0000   | オシレーター3はオフ。                              |
| CRD | A 5 | 1/010 0101   | ノイズ・ジェネレーターはオンで、相対周<br>波数は25。            |
| CRE | ХF  | X X X X 1111 | 音量が最大にセットされます。<br>バックグラウンド・カラーは共通で全文     |
| CRF | ØE  | 9999/1/119   | 字がブラック(の)でボーダー・カラーはダークブルー(6)、個々の文字はそれ自身の |
|     |     |              | 色で黒いバックグラウンドの上に表示されています (R=1)。           |

これらのレジスタ値は、23×22文字のビデオ・マトリックスのスクリーンを作り、個々の文字は黒のバックグラウンドの上に表示され、ビデオ・マトリックスを囲んでいるボーダー・エリアはダークブルーです。それにサウンド・ジェネレーター#2とホワイト・ノイズが出力します。

これら全部のレジスタにより異なった効果を作り出すことができます。

### 例:

CR∮の数を増すとビデオ・マトリックス領域は右へシフトします。CRBの数を減らすとオシレーター2の周波数はダウンします。CRFを%に変えるとビデオ・マトリックスのボーダーはダークブルーで残りますが、文字はブラックになり、バックグラウンドは別の異なった色に変わります。

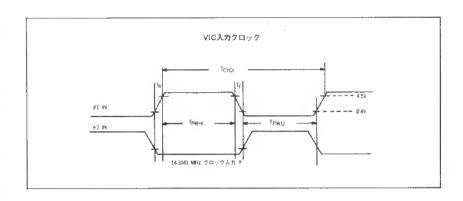
スクリーン上に絵を作るためには、VICにローとカラムの数および中心の値を 適切なレジスタにロードしなければなりません。

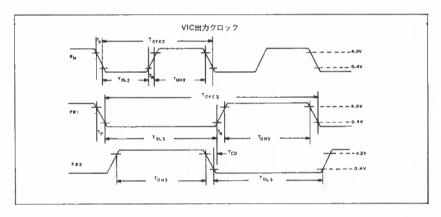
直流特性 Tagで~+56で、Vop-5V ±5%(他の指定がないとき)

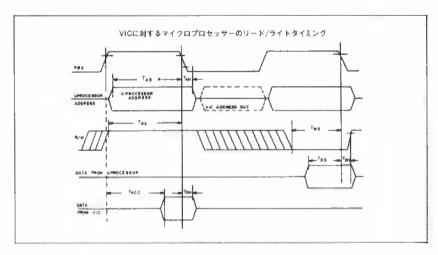
| パラメーター   | 最小                                       | 最大                        | タイプ             | 単位                                      |
|--|--|---------------------------|-----------------|---|
| リード/ライトリセット(オプション)<br>アドレスとデータ入力の状態<br>V <sub>IL</sub><br>V <sub>IH</sub><br>入力容量<br>入力溶量<br>入力漏れ(全出力はハイ・インピーダンス状態)  | -0.2<br>2.4                              | 0.4<br>5.6<br>8.0<br>10.0 | 5.0<br>1.0      | Volts<br>Volts<br>pF<br>µA              |
| アドレスとデータ出力状態<br>V <sub>O</sub> ,<br>V <sub>O</sub> ,<br>1 <sub>OL</sub> ーシンクカレント V <sub>OL</sub> ー 6.4<br>t <sub>OH</sub> ーソースカレント V <sub>OH</sub> ー 2.4<br>3状態のインピーダンス  | 2.4<br>2.4<br>260<br>1 x 10 <sup>8</sup> | 0.4                       |                 | Volts<br>Volts<br>mA<br>μΑ<br>Ohms      |
| クロック入力(タィとタッの入力)<br>周波数<br>容量<br>V <sub>IL</sub><br>V <sub>IR</sub>  | -0.2<br>4.5                              | 10.0<br>0.3               | 14.31818<br>5.0 | MHz<br>pF<br>Volts<br>Volts             |
| クロック出力(Pダ <sub>1</sub> ,Pダ <sub>2</sub> )<br>V <sub>CL</sub><br>(c. (* Ø.3 V olts V <sub>CL</sub><br>V <sub>GH</sub><br>(· v. 4.7 Volts V <sub>GH</sub><br>ローディング<br>間波数 | 1.6<br>V <sub>DD</sub> 2<br>200          | 0.3V<br>120.0             | 1.02            | Volts<br>mA<br>Volts<br>µA<br>pF<br>MHz |

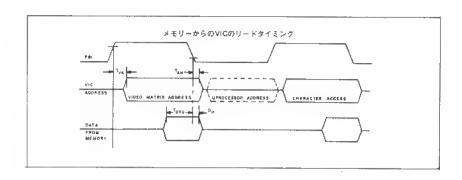
直流特性  $T_A \sim 9^{\circ}C \sim +50^{\circ}C, V_{DO} = 5V \pm 5\%$  (他の指定がないとき)

| パラメーター   | 最小                  | 最大   | タイプ  | 单位    |
|--|---------------------|------|------|-------|
| 音の構成   |                     |      |      |       |
| 出力インビーダンス                                      |                     | 2000 | 1000 | Ω     |
| <b>最大カレント(シンクまたはソース)</b>                       |                     | 500  |      | μĀ    |
| 出力オフセットボルテージ                                   | 2.2                 | 2.8  | 2.5  | Volts |
| V <sub>DH</sub> (最大振幅)                         | 3.2                 |      | 3.5  | Volts |
| You (最大振幅)                                     |                     | 1.8  | 1.5  | Volts |
| V <sub>0H</sub> (最小振幅)                         | 2.55                |      | 2.6  | Volts |
| V <sub>0L</sub> (最小振幅)                         |                     | 2.45 | 2.4  | Volts |
| Pot入力  |                     |      |      |       |
| V <sub>TRIGGER</sub> (ライジングエッジ)                | 2.2                 | 2.8  | 2.5  | Volts |
| Potリセット  |                     | 6.5  |      | 14.15 |
| $V_{0L}$ $I_{C_3}$ (a. $V_{C_3} = \emptyset.2$ | 500                 | 0.2  |      | Volts |
| Co. (c) VOL - 0.2                              | 500                 |      |      | μА    |
| ライトペン入力(オプション)                                 |                     |      |      |       |
| V <sub>TROGER</sub> (フォーリングエッチ)                | 2.8                 | 2.2  | 2.5  | Volts |
| ∮M(オプション)                                      |                     |      |      |       |
| V <sub>Ot</sub>                                |                     | 0.4  |      | Volts |
| lor (≈ Ø.3 Volts V <sub>OL</sub>               | 1.6                 |      |      | mA.   |
| <b>K</b> a⊲                                    | V <sub>DO</sub> =.7 |      |      | Volts |
| I <sub>OH</sub> (r) 4.7 Volts V <sub>OH</sub>  | 100                 |      | l    | μA    |
| ローディング   |                     | 60   |      | pΕ    |
| 剛波散  |                     |      | 2.04 | MHz   |
| バスの使用(オブション)                                   |                     |      |      |       |
| Y <sub>CR</sub>                                |                     | 0.3  |      | Volts |
| kox.   | 1.6                 |      |      | mA    |
| V <sub>LM</sub>                                | 2.4                 |      | [    | Volts |
| low  | 100                 |      |      | µA.   |
| Y <sub>LKI</sub>                               | 4.75                | 5.25 | 5.00 | Volts |
| l <sub>litt</sub>                              |                     | 150  | 120  | mA    |







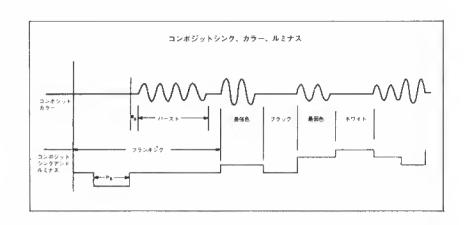


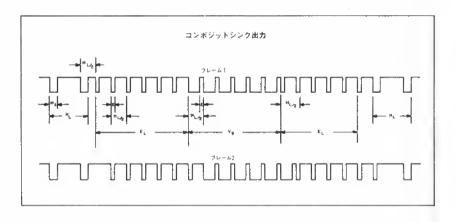
### VICシステムタイミング

| 記 号                              | 特性                                      | 最小        | 標準    | 最大    | 単位       |
|----------------------------------|---|-----------|-------|-------|----------|
|                                  | VIC入力クロックタイミング                          | "         |       |       |          |
| T <sub>CYC1</sub>                | 入力クロックサイクルタイム                           | 69.82     |       | 69.84 | R5       |
| T <sub>PWH1</sub>                | "H"クロック                                 | 20        |       | İ     | ns       |
| T <sub>PWL1</sub>                | "ピクロック                                  | 20        |       |       | ns       |
| T <sub>R</sub> , T <sub>F</sub>  | 立上がり立下がり時間                              |           |       | 10    | ns       |
|                                  | VIC出力クロックタイミング                          |           |       |       |          |
| T <sub>CYC2</sub>                | 2MHはクロックサイクルタイム                         | 480       |       | 500   | ns       |
| T <sub>OL2</sub>                 | "L"ヶMクロック出力                             | 200       |       | 260   | ns       |
| T <sub>OH2</sub>                 | "H"∮Mクロック出力                             | 180       |       | 250   | DS.      |
| T <sub>CYC</sub> 3               | IMHzマイクロプロセッサークロックサイクルタイム               | 960       | 1     | 990   | กร       |
| TOLI                             | "L"P#IP#27 a y 7                        | 380       |       | 500   | DS       |
| TOHB                             | "H"PタiPクi2ロック                           | 380       | i     | 500   | ns       |
| T <sub>CD</sub>                  | 4Vでのクロック選録時間                            | 5         |       | 20    | DS.      |
| T <sub>R</sub>                   | 最大Qでの立上がり時間                             |           |       | 80    | ns       |
| T <sub>F</sub>                   | 最大Cでの立下がり時間                             |           |       | 40    | ns       |
|                                  | VICに対するマイクロプロセッサーのリード/ライトタイミング          |           |       |       | -        |
| T <sub>AS</sub>                  | アドレスセットアップ時間                            | 375       |       |       | ns       |
| TAH                              | アドレス・ホールド時間                             | 5         |       |       | ns ns    |
| TRS                              | <b>胚込みセットアップ時間</b>                      | 375       |       | ļ     | F15      |
| Tuys                             | 書込みセットアップ時間                             | 275       |       |       | ns       |
| Tos                              | データセットアップ時間                             | 200       |       |       |          |
| TACC                             | データアクセス時間                               | 350       |       |       | □ □5     |
| T <sub>DH</sub>                  | データホールド時間                               | 30        |       |       | ns<br>ns |
|                                  | メモリーからのVICのリードタイミング                     |           |       |       | 113      |
| Tva                              | Pがから有効アドレスまでの時間                         | 1         |       |       |          |
| TAH                              | アドレスホールド時間                              | 10        | 1     |       | ns       |
| Togi                             | テータセットアップ時間                             | 60        |       |       | ns       |
| D <sub>H</sub>                   | データホールド時間                               | 20        |       |       | ns       |
| <u> Н</u>                        | *************************************** | 20        | -     |       | ns       |
| BLANKING.                        | 色と発光タイミングの同期構成<br>ブランキング順間(ビデオなし)       | 10.0      | 11.0  | 12.0  |          |
| B <sub>c</sub>                   | ブリーズウェイ                                 |           |       | 12.0  | μS       |
| BURST                            | カラーバーストリファレンス信号                         | .3<br>4.0 | .5    | -7    | jaš      |
| BOK31                            |   | 4.0       | 5.0   | 6.0   | #5       |
| H <sub>s</sub>                   | 出力タイミングの同期構成<br>水平同期パルス                 |           |       |       |          |
| H <sub>i</sub>                   | 水平ウイン時間                                 | 4.0       | 5.0   | 6.0   | μ5       |
|                                  | 水平フィン時間<br>  1/2 水平ライン時間                | 63.0      | 63.5  | 64.0  | μεS      |
| H <sub>L/2</sub><br>F            |   | 30.0      | 31.5  | 32.5  | £1.5     |
| -                                | イコーリセーション・バルス                           | 2.0       | 2.5   | 3.0   | 215      |
| fi                               | イコーリセーション時間                             | 188.0     | 190.5 | 192.0 | μ5       |
| V <sub>h</sub>                   | 多画問期バルス                                 | 188.0     | 190.5 | 192.0 | µ5       |
| V <sub>5</sub> to V <sub>5</sub> | 事画面期間の時間                                |           | 16.66 |       | FF15     |

<sup>」。</sup> ・ カラーバースト保存は 3,6次約前州収のカラー・フェイスで 何のカラ・検 利はまれにより前定されます。 もとよば 看傷度のフル は1,5次分が同様でき、バースト情報が同<sub>年</sub>の間もあ

あるら、LYousの場所ももらます。 2、Vg/間側の4.開閉の開節は 混合モーシで、202.5でも、 3、Vg/間側が4.開間が4個間は お混合モートで フェ ムボスリ202です も、NTSCOのよ





# MPS6522汎用インターフェイスアダプタ

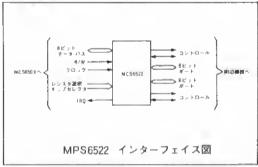
# 概要

MPS 6522汎用インターフェイスアダプタ (Versatile Interface Adapter: VIA) は、MPS 652Øの機能をすべて備えており、さらに2つの強力なインターバル・タイマー・シリアルデータの送受信ができるシフト・レジスタ、そしてペリフェラル・ポートの入力データラッチが加わっています。強化されたハンドシェイク能力により、マルチプロセッサシステムでのVIA相互の双方向データ転送が可能です。

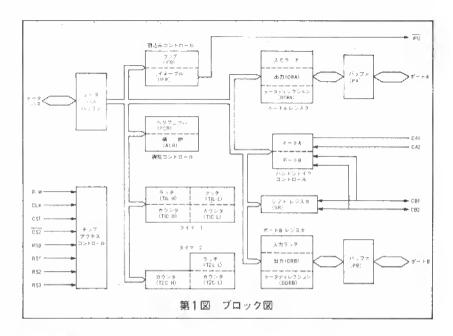
周辺機器の制御は主に2つの8ビット双方向ポートによっておこないます。これらのポートは入力にも出力にも使うことができます。また、数本の1/O線をインターバル・タイマでコントロールすることができ、これによって方形波を発生させてその周波数をプログラムで制御したり、外部パルスをカウントしたりすることができます。このチップの多くの強力な機能を容易に発揮できるように、チップ内部のレジスタとして、割込みプラグレジスタ、割込みイネーブルレジスタ、および2つの機能コントロールレジスタがあります。

### 特長

- ●MPS6520をよりパワフルに強化
- N チャンネル デブレッションロードによる 5 V 単一電源
- ●完全スタチック動作、TTLコンパチプル
- CMOSコンパチプルなペリフェラルコント ロール線
- ●プロセッサと周辺機器とのデータ転送を強力 にコントロールできる充実したハンドシェイ ク機能。







# プロセッサインターフェイス

ここでは、MPS 6522とプロセッサとのインターフェイスに使うバスとコントロール用の説明をします。このインターフェイスのAC、DC特性は最後の特性表に記してあります。

# フェーズ 2 クロック(∮2)

MPS6522とプロセッサとの間のデータのやりとりは、フェーズ 2 クロック  $^{\circ}$  H '' の時のみおこなわれます。さらにこの  $\phi$  2 は、チップの中のタイマやシフトレジスタのクロックとしても使われます。

# 2. チップセレクト(CS1、CS2)

2本のチップセレクト入力は、プロセッサのアドレスパスから直接またはデコーダを介してつながります。CS1  $^{\circ}H''$   $^{\circ}CS2$   $^{\circ}L''$  の時、MPS6522 のある選択されたレジスタがアクセスされます。

# 3. レジスタ選択線(RSØ、RS1、RS2、RS3)

4本のレジスタ選択線はプロセッサのアドレスバスにつなぎ、プロセッサはこれによりアクセスするレジスタを選択します。16通りの組合せでアクセスされるレジスタは次の通りです。

| RS3 | RS2 | R\$1 | RSØ | レジスタ            | 注                           | RS3 | R\$2 | RSI | RSØ | レジスタ           | 注                           |
|-----|-----|------|-----|-----------------|-----------------------------|-----|------|-----|-----|----------------|-----------------------------|
| L   | L   | L    | L   | ORB             |                             | Н   | L    | L   | L   | T2L-L<br>T2C-L | カウンタ読み出し                    |
| L   | Ł   |      | Н   | , ORA           | ハンドシェイクの<br>コントロール          | Н   | L    | L   | Н   | 12C- H         | T2L-LからT2C-Lへの<br>データ転送のトリガ |
| L   | L   | н    | L   | DDRB            |                             | н   | L    | н   | L   | SP             |                             |
| L   | L   | н    | н   | DDRA            |                             | н   | L    | н   | Н   | ACR            |                             |
| L   | Н   | L    | L   | 1111-L<br>TIC-L | ラッチへ書き込み<br>カウンタ読み出し        | н   | Н    | L   | L   | PCR            |                             |
| L   | Н   | L.   | Н   | т1с н           | Tit-LからTit-Lへの<br>データ転送のトリガ | Н   | Н    | L   | Н   | . IŁK          |                             |
| Ł.  | Н   | Н    | L   | T1L-L           |                             | Н   | Н    | Н   | L   | (ER            |                             |
|     | н   | H    | н   | TIL-H           | -                           | Н   | Н    | н   | н   | ORA            | ハンドシェイクに<br>影響せず            |

### 4. リード/ライト線(R/W)

MPS6522とプロセッサとの間のデータのやりとりの方向は、このR/Wで決まります。R/Wが  $^{\circ}L''$  の時、データはプロセッサからMPS6522の選ばれたレジスタへ転送され (書き込み動作)、 $^{\circ}H''$  の時はそのレジスタから外部へ転送されます (読み出し動作)。

### 5 . データバス(DBØ-DB7)

MPS6522とプロセッサとのデータ転送はこの8ビット双方向データバスでおこないます。パスドライバーは、通常は高インピーダンス状態になっていて、チップセレクト信号が来て(CS1=H、CS2=L)、R/W= $^{\circ}$ H"になると、データバス上のデータが選択されているレジスタに送り込みます。

# 6. リセット(RES)

リセット入力はT1、T2、SRを除くすべてのレジスタの内容を貸にします。 これにより周辺機器とのインターフェイス線は入力状態になり、タイマ、シフトレジスタ等はディスエーブルされ、このチップからの割込みはディスエーブルとなります。

# 7、割込みリクエスト(IRQ)

割込みリクエスト出力は、内部の割込みプラグがセットされ、その時対応する割込みイネーブルビットが1ならば  $^*L''$  になります。この出力はオープンドレインになっているのでシステム内の他の割込みリクエストとワイアードORをとることができます。

# ペリフェラルインターフェイス

ここでは、MPS6522の内部レジスタが周辺機器をドライブする時に使うバスライン、コントロールラインを簡単に説明します。

### 1. Aポート(PAG-PA7)

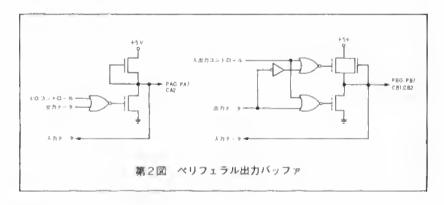
Aポートには8本の線があって、入出力の切替はデータ・ディレクション・レジスタをプログラムすることによって1ビットずつ個別におこなえます。出力ピンのデータは出力レジスタから送られ、入力ピンのデータはCA1のコントロールによって内部のレジスタにラッチされます。こういった動作はすべてプロセッサが、チップ内部の制御レジスタを介してコントロールします。PAØ-PA7は、入力時には標準TTLファンイン1、出力時にはファンアウト1です。

### 2. Aポートコントロール線(CA1、CA2)

Aポートのコントロール線は2本あって、割込み入力およびハンドシェイク出力として動作します。それぞれ対応する割込みイネーブルビットと共に割込みフラグをコントロールします。さらにCA1はAポートの入力データのラッチをコントロールします。動作モードの切替は、プロセッサがチップ内部のコントロールレジスタを介しておこないます。CA1は高インピーダンスの入力のみの線ですが、CA2は標準TTLのファンイン1、ファンアウト1です。

### 3. Bポート(PBØ-PB7)

Bポートも8本の双方向データ線で、出力レジスタとデータ・ディレクション・レジスタにより、Aポートとほぼ同様にコントロールされます。さらにこのポー



トでは、2つのインターバルタイマの片方の出力をPB7に出力し、もう一つのタイマでPB6に入力するパルスを計数することができます。この8本は標準TTLでファンイン1、ファンアウト1で、出力モード時には1.5Vで30mAのソース能力がありますので、ダーリントンのトランジスタスイッチなどを直接ドライブすることができます。

### 4. Bポートコントロール線(CB1、CB2)

Bポートのコントロール線も割込み入力およびハンドシェイク出力として動作します。CA1、CA2と同様、それぞれ対応する割込みイネーブルビットと共に割込みフラグをコントロールします。またシフトレジスタでコントロールするシリアルポートにもなります。標準TTLでファンイン1、ファンアウト1で、ダーリントランジスタスイッチを直接ドライブできるよう、出力時に1.5Vで1mAのソース能力を持っています。

# MPS6522の動作

ここでは第1図に示したいろいろなロジックと、MPS6522の内部の動作について詳しく説明します。

# A. $\vec{r}$ — $\phi$ // $\vec{r}$ /

これらのパッファの電流、電圧のドライブ能力は前の章で述べた通りです。AC、DC特性は最後の特性表に記してあります。

# B. チップアクセスコントロール

チップアクセスコントロールは、チップセレクト9条件を検出し、レジスタセレクト入力をデコードして選択されたレジスタをアクセスするロジックで、データのやりとりの方向とタイミングを決定するR/Wと必も含まれます。MPS6522にデータを書き込む時は、まずデータは約2でデータ入力レジスタにラッチされ、次に約2・チップセレクトで選択された内部レジスタへ転送されます。こうすることによってペリフェラルI/Oライン上のデータをグリッチなしに変化させることができます。プロセッサがMPS6522から読み出す時は、データは約2で選択されたレジスタから直接データバスに乗ります。

# C. ポートA レジスタ群、ポートB レジスタ群

8 ビットのペリフェラルポートをアクセスするのに、各々3 つずつのレジスタを使います。まず、データ・ディレクション・レジスタ (DDRA、DDRB) は、ペリフェラルピンを入力にするか出力にするかを決定します。このレジスタのあるビットが g ならば対応するピンは入力となり、1 ならば出力となります。

次に出力レジスタ (ORA、ORB) と人力レジスタ (IRA、IRB) があります。ピンが出力にプログラムされていれば、出力レジスタの対応するビットが 1 ならばピンは  $^{\circ}$  11'' に、 $^{\circ}$   $^{\circ}$  ならば  $^{\circ}$   $^{$ 

ペリフェラルポートから読み込んだデータは入力レジスタ (IRA、IRB) に入り、そこから直接データバスへ送り出すことができます。人力ラッチ動作をディスエーブルするとIRAは常にPAのピンと同じ内容になります。これをイネーブルするとCAIの割込みフラグ (IFR1) がセットされる直前のポートAの内容がIRAに入ります。

IRBレジスタも大体似たような動作をしますが、IRAと違うのは、IRBの出力にプログラムされているビットの内容は、ピンの電圧ではなくORBの当該ビットのデータが入ります。これによってプロセッサはIRBを読む時に、ポートBの電圧が充分に振れていなくても(ポートBは1.5Vで30mAのソース能力がある)正しいデータが読み取れるわけです。ポートBの入力ラッチがイネーブルの時CB1の割込みフラグがセットされると、IRBは上記の組合せのデータを割込みフラグがクリアされるまでラッチします。

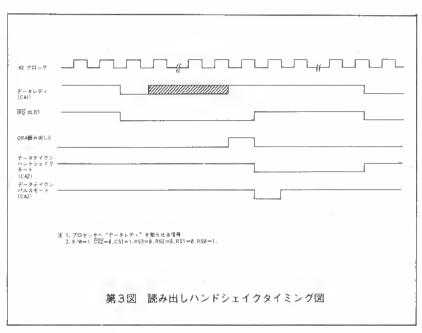
### D. ハンドシェイクコントロール

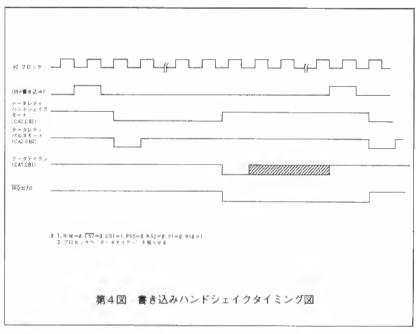
MPS6522は、プロセッサと周辺機器とのデータ転送をハンドシェイクラインの動作によって強力にコントロールします。ポートAのライン (CA1、CA2) は読み出し、書き込み両方のハンドシェイクをおこないますが、ポートBのラインは書き込みのハンドシェイクのみをおこないます。

#### D-1. 読み出しハンドシェイク

読み出しハンドシェイクは、プロセッサが周辺機器からデータを読み出すさいのコントロールを効率よくおこないます。まず周辺機器は〝データ・レディ〞信号を出して、ペリフェラルボートにデータが揃っていることをプロセッサに知らせます。プロセッサはこの信号により割込みを起してデータを読み込み、データ・テイクン゜信号を返します。周辺機器はこの信号に対し次のデータ・レディで応答し、データ転送が終るまでこのやりとりを続けます。

MPS6522では、読み出しの自動ハンドシェイクがおこなえるのはポートAだけです。CA1がデータ・レディ信号を受取り、CA2でデータテイクン信号を返します。データ・レディ信号は内部フラグをセットし、プロセッサはこれによって制込みを起すか、またはソフトウェアでボーリングをして検出します。データ・テイクン信号は単発のパルスで出力することも、DCレベルで出力することもできます。DCレベルで出力する場合は、プロセッサが「L"にセットし、次のデータ・レディ信号がこれをクリアします。この動作を第3図に示します。





### D-2. 書き込みハンドシェイク

プロセッサが周辺機器にデータを書き込む書き込みハンドシェイクの手順は、 前述の読み出しハンドシェイクとよく似ていますが、今度はプロセッサがMCS 6522を介してデータ・レディ信号を出し、周辺機器がデータ・テイクン信号で応 答します。この動作はポートAでもポートBでもおこなえます。CA2、CB2はPCレベルまたはパルスでデータ・レディ信号を出し、CA1、CB1が周辺機器からデータ・テイクン信号を受取って割込みフラグをセットし、データ・レディをクリアします。この手順を第4図に示します。

# タイマ

### E-1. タイマ1

インターパルタイマT1は、2個の8ビットのラッチと16ビットのカウンタで構成されています。ラッチには、カウンタにロードするデータが入ります。カウンタはデータがロードされると、フェーズ2クロックによってカウントダウンを始め、ゼロになると割込みフラグをセットして $\overline{IRQ}$ を  $\overline{IRQ}$ を  $\overline{IRQ}$  にします。この後タイマはその動作モードによって、それ以上の割込みをディスエーブルするか、またはラッチの内容を再び受取ってカウントダウンを続けるかします。さらにカウンタがゼロになるたびにペリフェラルピンPB7の出力を反転させることもできます。以下にタイマの動作の各モードについて詳しく述べます。

### E-2. タイマ1のレジスタへの書き込み

T1の四つのアドレスへデータを書き込むと、次のような動作をします。

| R\$3 | RS2 | #81 | ĦSØ | 動 作 (R/W=L)  |
|------|-----|-----|-----|--|
| Ł    | Н   | L   | L   | ラッチ下位8ビットへの書き込み  |
| L    | н   | L   | н   | ラッチ上位へ書き込み カウンタ上位へ書き込み<br>ラッチ下位をカウンタ下位へ転送 TT割込みブラグリセット |
| L    | н   | н   | L   | ラッチ下位へ書き込み   |
| L    | н   | н   | н   | ラッチ上位へ書き込み TI 削込みフラグリセット                               |

プロセッサはカウンタの下位8ビット (TIC-L) に直接書き込むことはできません。カウンタ上位への書き込みをおこなった時にラッチ下位のデータが自動的に転送されるようになっています。タイミング動作はカウンタ上位が書き込まれてから開始されるので、カウンタ下位への直接書き込みは必要ありません。

アドレスセットの後半の2つは、進行中のカウントダウン動作に影響を与えず にラッチに書き込むためのアドレスです。これについては後で詳しく述べます。

### E-3. タイマ1のレジスタの読み出し

読み出し時のタイマ1の四つのアドレスの動作は次の通りです。

| RS3 | R\$2 | R\$1 | RSØ | 1t) /F (R/W−H)                |
|-----|------|------|-----|-------------------------------|
| L   | н    | L    | L   | TIカウンタ下位8ビット読み出し TI割込みフラクリセット |
| L   | Н    | L    | н   | 上位カウンタ読み出し                    |
| L   | Н    | Н    | L   | 下位ラッチ読み出し                     |
| L   | Н    | Н    | н   | 上位ラッチ読み出し                     |

### E-4. タイマ1の動作モード

タイマ1の動作モードは補助コントロールレジスタの2つのビットが決定します。これらのビットの状態とその時の動作モードは次の通りです。

| 出力コントロール ACR7 | 79-35 ACR6 | €-ド                   |  |
|---------------|------------|-----------------------|--|
| Ø             | p          | ワンショット、PB7出力ディスエープル   |  |
| Ø             | 1          | ブリーランニング、PB7出力ディスエーブル |  |
| 1             | g          | ワンショット、PB7出力イネーブル     |  |
| 1             | 1          | フリーランニング、PB7出力イネーブル   |  |

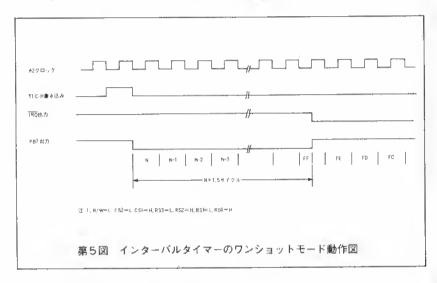
#### E-5. タイマ1ワンショットモード

インターバルタイマのワンショットモードでは、タイマへデータをロードするたびに単発の割込みがおこります。他のインターバルタイマと同様、T1C-Hレジスタへの書き込みから割込み発生までの遅れ時間はタイミングカウンタにロードされたデータによって決まります。タイマ1は割込みを起こすだけでなく、ペリフェラルピンPB7に単発のネガティブパルスを出力することができます。出力イネーブル状態(ACR7=1)でT1C-Hへデータを書き込むとPB7は、L″になり、タイマ1のタイムアウトで、H″に戻ります。

〈注〉PB7はDDRB7とACR7の両方によって出力モードに指定されますが、どちらも1になっていると、タイマ1がPB7をコントロールし、ORB7は無効となります。

ワンショットモードでは上位ラッチ1の書き込みは動作に何の影響も与えませんが、下位ラッチへはタイマがTIC-Hへの書き込み操作によってカウントダウンを始める前に正しいデータを入れておく必要があります。プロセッサが上位カウンタへデータを書き込むと、T1の割込みフラグがクリアされ、下位ラッチの内容が下位カウンタへ送られ、タイマはクロック周波数でカウントダウンを始めます。出力イネーブルになっていればこの時PB7はプロセッサによる書き込みの次のフェーズ2で「L"となります。カウンタの内容がゼロになると、T1の割込

みフラグがセットされ、割込みイネーブルならば $\Pi Q$ が  $^*L''$  になり、PB7は $^*L''$  になりPB7は $^*H''$  に戻ります。この時もクロックによるカウントダウンは続いていて、プロセッサがカウンタの内容を読んで割込み発生からの経過時間を知ることができるようになっています。しかしT1割込みフラグは、割込みコントロールの項で述べるように、クリアされていなければ再度セットされることはありません。ワンショットモードのタイミング図を第5図に示します。

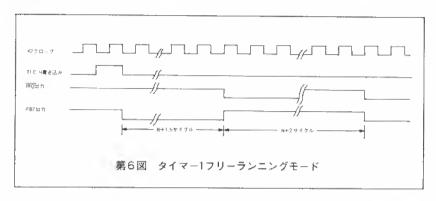


### E-6. タイマ1 フリーランニングモード

T1のラッチによる最大の特長は、フリーランニングモードにした時に一定の間隔で連続して割込みをかけることができ、PB7にプロセッサの割込み応答時間の変動によって影響を受けない一定周波数の方形波を出すことができることです。フリーランニングモード (ACR6=1) では、カウンタがゼロになる毎に割込みフラグがセットされ、PB7の信号が反転します。ゼロになった後、タイマはラッチの内容16ビットをカウンタへ転送し、そこからカウントダウンを始めます。割込みフラグは、T1C-Hへの書き込みか、T1C-Lの読み出しか、後迷するフラグへの直接書き込みによってクリアされ、次のタイムアウトの時フラグがセットされるようタイマを書き直す必要はありません。

MCS6500ファミリのインターバルタイマはすべて再トリガが可能です。そしてカウンタの内容を書き替えると、必ずタイムアウトまでの周期も新しくイニシャライズされます。実際にタイマがゼロになる前にプロセッサがタイマの上位カウンタ (TIC-H) の書き替えを続ければ、タイムアウトは決して起りません。しかしラッチへの書き込みは進行中のダウンカウントに影響を与えず、ラッチへ書き

込まれたデータは次のタイムアウトの周期となります。この機能は出力イネーブルの時、特に効果を発揮します。この時はタイムアウトのたびにPB7の信号が反転し、割込みフラグがセットされるので、割込みのたびに新しいデータをラッチに入れて次の半サイクルの周期を決定してやることによってPB7に複雑な波形を発生させることができます。フリーランニングモードのタイミング図を第6図に示します。



### F. タイマ2

タイマ2には、ワンショットモードのみのインターバルタイマとしての機能と、ペリフェラルピンPB6に入るネガティブパルスを計数する機能とがあって、補助コントロールレジスタACRのうちの1ビットで切替えます。このタイマは書き込み可能な下位ラッチ(T2L-L)読み出し可能な下位カウンタおよび読み出しと書き込み可能な上位カウンタで構成されています。カウンタレジスタは、∮2でダウンカウントする16ビットカウンタとして働きます。タイマ2の各レジスタのアドレスは次の通りです。

| RS3 | RS2 | RSI | RSØ | R/W=0                                 | RW=1                     |
|-----|-----|-----|-----|---------------------------------------|--------------------------|
| Н   | L   | L   | L   | T2L-Lへ書き込み                            | †20- Li放み出し<br>創込みフラグクリア |
| н   | L   | L   | н   | T2C-Hへ書き込みT2L-LをT2C-Lへ<br>転送割込みフラグクリア | T2C=H級み出し                |

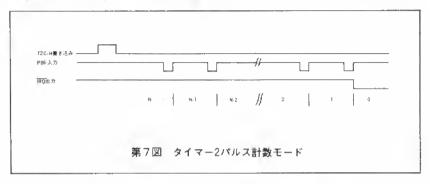
# F-1. タイマ2 インターバルタイマモード

インターバルタイマとしては、T2はワンショットモードでT1と同じように働きます。このモードでは、T2C-Hへの書き込みのたびに単発の割込みが発生します。タイムアウト後もカウンタはダウンカウントを続けますが、割込みフラグのセットは、ゼロを過ぎたカウンタが再びセットしないようにディスエーブルされます。これをイネーブルするにはプロセッサはT2C-Hを書き直さねばなりません。割込みフラグはT2C-Cの読み出しかT2C-Hへの書き込みでクリアされます。

この動作のタイミング図は第5図です。

### F-2、タイマ2パルス計数モード

パルス計数モードでは、T2はあらかじめ決められた数のPB6に入るネガティブパルスを計数します。このため、まずT2に数をロードします。T2C-Hへの書き込みで割込みパルスはクリアされ、カウンタはPB6にパルスが入るたびに減少を始めます。T2がゼロになると割込みフラグがセットされますが、カウンタはさらにPB6のパルスで減少を続けます。次のタイムアウトで割込みフラグをセットさせるためには、T2C-Hを書き替えねばなりません。このモードのタイミングを第7図に示します。パルスはp2の立ちあがりで  $^{8}$ L  $^{6}$  でなければなりません。



### G.シフト・レジスタ

シフトレジスタは内部の8進カウンタにより、CB2ピンを通じてシリアルデータの送受信をおこないます。シフトパルスは外部からCB1ピンに加えることもできますし、動作モードによっては内部で発生させてCB1ピンに出力し、外部機器でのシフトをコントロールすることもできます。シフトレジスタの動作モードは補助レジスタのビットが決定します。プロセッサはこれらのビットを介して動作モードを決定します。

### G-1. シフトレジスタ入力モード

補助コントロールレジスタのビット4は入出力の切替えビットです。シフトレジスタの動作モードは、シフトパルスの発生源によって入力モードに3種類、出力モードに4種類あります。ACR4=Øの入力モードは、ACR3とACR2によって次のように決定されます。

| ACR4 | ACR4 ACR3 A |   | € - ド            |  |
|------|-------------|---|------------------|--|
| 0    | p           | B | シフトレジスタ、ディスエーブル  |  |
| ø    | В           | 1 | タイマ2によるシフト入力     |  |
| 9    | 1           | 0 | システムクロックによるシフト入力 |  |
| ıl . | 1           | 1 | 外継パルスによるシフト入力    |  |

### G-2. モードØØØシフトレジスタ ディスエーブル

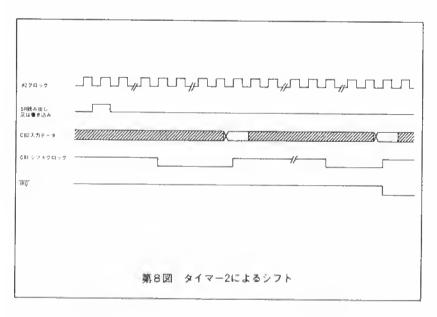
999年ードではシフトレジスタはディスエーブルされます。この時プロセッサはシフトレジスタに対し読み出し、書き込みをおこなうことはできますが、シフト動作はディスエーブルされており、CB1とCB2の動作はペリフェラルコントロールレジスタ (PCR) の該当ビットで決まります。

また、シフトレジスタ割込みフラグもディスエーブルされます。

### G-3、モード001タイマ2によるシフト入力

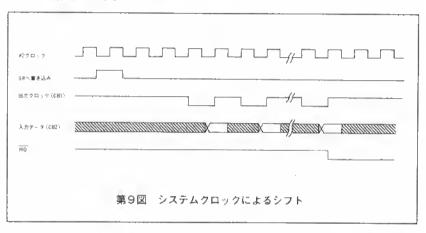
このモードではシフト速度はT2の下位8ビットでコントロールされ、シフトパルスはCB1ピンに出力されて、外部機器でのシフトの制御に使われます。このシフトパルスの周期は、システムロックとT2のラッチ下位8ビットの内容で決まります。

シフト動作はシフトレジスタへの読み出し、書き込みによってトリガされます。データはまずSRの下位ビットに入れられ、クロックパルスの後縁(CB1の立ち上がり)でとなりの上位ビットにシフトされます。第8図に示すようにクロックパルスの先端(CB1の立ち上がり)が来る後に次のデータが来ていなければなりません。このデータはクロックパルスの後縁の次のシクテムクロックサイクルでシフトレジスタに読み込まれます。そしてクロックバルスが8つ入ると、シフトレジスタの割込みフリグがセットされ、IRQは『L』になります。



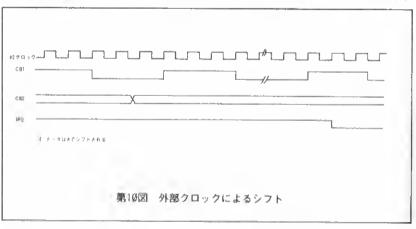
# G-4. モードØ1Øシステムクロックによるシフト入力

このモードではシフトはシステムクロックの周波数でおこなわれます。CB1はシフトパルスを出力し、外部機器をコントロールします。タイマ2はインターバルタイマとして働き、シフトレジスタの動作には関係しません。シフトレジスタへ読み出しまたは書き込みをおこなうとシフト動作が始まり、データはまずビットりに入り、クロックパルスの後縁で順に高次ビットへ送られます。8クロックパルス経過後、シフトレジスタの割込みフラグがセットされ、CB1のクロックパルス出力は停まります。



### G-5. モードØ11外部クロックによるシフト入力

このモードではCB1はシフトパルスの入力ピンとなって、外部機器が自身の決める転送速度でデータをシフトレジスタに入れることができるようになります。



シフトレジスタのカウンタは8ビット読み込むごとにプロセッサに割込みをかけますが、その後もパルスカウンタとして働き、シフト動作は続きます。シフトレジスタへ読み出しまたは書き込みをおこなうと割込みフラグはリセットされ、シフトレジスタカウンタはイニシャライズされて次の8ビットをかぞえはじめます。データは、CB1のシフトパルスの先端の次のシステムクロックサイクルでシフトされます。このため、CB1が ドH ″になった後システムクロック 1 サイクルの間はデータを保持しなければなりません。動作のタイミングを第10図に示します。

### G-6. シフトレジスタ出力モード

入出力コントロールビット(ACR4)を1にするとシフトレジスタは出力となり、ACR3とACR2の組合せにより四つのモードが選べます。どのモードでも、シフトレジスタのビット7の内容がCB2ピンに出力され、同時にビットダへシフトバックします。人力モードと同様、CB1はシフトパルスを出力することも、外部から入力することもできます。

4つのモードは次のとうりです。

| ACR4 | ACR3 ACR2 |      | €− κ                    |  |  |
|------|-----------|------|-------------------------|--|--|
| 1    | 8         | ø    | T2によるフリーランニングモードのシフト出力  |  |  |
| 1    | ß         | 1    | T2によるシフト出力、CBIにシフトパルス発生 |  |  |
| 1    | 1         | . 18 | システムクロックによるシフト出力        |  |  |
| 1    | 1         | 1    | 外部パルスによるシフト出力           |  |  |

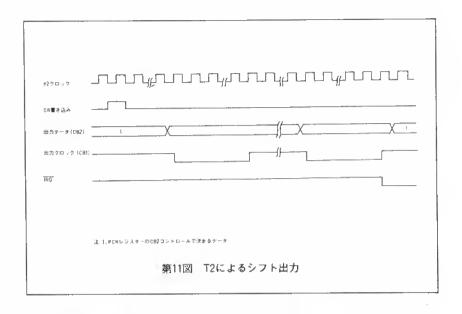
### G-7. モード100プリーランニング出力

このモードのシフト速度はモード1 $\emptyset$ 1と同様T2が決定します。モード1 $\emptyset$ 1と違うのは、このモードではSRカウンタによってシフト動作が止められることがないという点です。シフトレジスタのビット7 (SR7) はビット $\emptyset$ に戻されますので、ロードされているデータ8ビットはクロックに従ってCB2に何回でも現われます。このモードではシフトレジスタカウンタはディスエーブルされています。

### G-8. モード16/1T2によるシフト出力

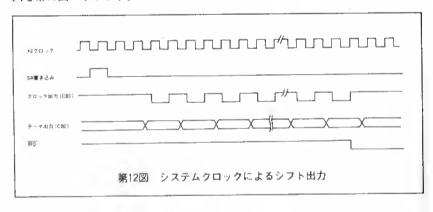
このモードでは前のモードを同様シフト速度はT2で決まります。しかし今度はシフトレジスタへの書き込みまたは読み出しのたびにSRカウンタがリセットされ、データ8ビットがシフトされてCB2に送り出されます。同時にシフトパルスを8発CB1に発生し、外部機器でのシフトをコントロールします。8発のシフトパルスの後はシフト動作は止まり、CB2はペリフェラルコントロールレジスタのCB2コントロールビット (PC5) がコントロールするようになります。

最後のタイムアウトの前にシフトレジスタをロードしなおせばシフト動作は続きます。タイミング図を第11図に示します。



### G-9. モード110システムクロックによるシフト出力

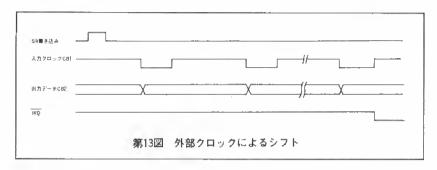
このモードの動作は第11図と似ていますが、シフト速度は $\phi$ 2ピンのシステムクロックになり、T2はインターバルタイマとしての動作に戻ります。タイミング図を第12図に示します。



# G-10. モード111 外部パルスによるシフト出力

このモードではシフトは外部機器がCBJピンに加えるパルスによっておこなわれます。SRカウンタはパルスが8発入るたびにSR割込みフラグをセットしますが、その後もシフトは続きます。プロセッサが読み出しか書き込みをおこなう

たびにSR割込みフラグはリセットされ、SRカウンタはイニシャライズされて、CB1のシフトパルスのカウントを始めます。シフトパルスが8発来ると割込みフラグがセットされるので、この時プロセッサは次のデータ1バイトをロードすることができます。



# 割込みコントロール

### H. 割込みコントロール

MPS6522内部の割込みコントロールには、割込みフラグを立てること、割込みをイネーブルすること、プロセッサに割込みが発生したことを伝えることの3つの動作があります。割込みフラグは、チップ内部で割込み条件が満たされるか、チップに入力があるとセットされ、サービスがおこなわれるまでセットされています。割込みの発生源を知るためにはプロセッサは、フラグレジスタをアキュムレータに読み込んで、左か右にシフトして条件プランチ命令を実行し、割込みフラグを優位順位の高い方から調べます。

割込みフラグは、割込みイネーブルビットと対になっています。このビットはプロセッサからセット / リセットされ、対応するフラグのプロセッサへの割込みをイネーブルします。割込みの発生によって割込みフラグが1にセットされて、その時相手の割込みイネーブルビットが1になっていれば $\overline{IRQ}$ は  $^{\circ}L$  "になります。 $\overline{IRQ}$ はオープンコレクタになっていて、他のデバイスからの割込みリクエストとワイヤードORを収ることができます。

MPS6522では、割込みフラグは1つのレジスタに納められていて、そのレジス

| Eyk            | 7                 | 6  | 5  | 4   | 3   | 2  | 1   | 0   |
|----------------|-------------------|----|----|-----|-----|----|-----|-----|
| 耐込みフラブ<br>レジスタ | IRQ               | τi | Т2 | ĆB1 | CB2 | SR | CAT | CA2 |
|                | セット/クリア<br>コントロール | T1 | T2 | CB1 | CB2 | ŚR | CAT | GA2 |

タの第7ビットはチップ内で割込みが発生すると1になるようになっています。 これによって割込み発生源が分散しているシステムでのポーリングが容易におこ なえます。

# H-1. 割込みフラグレジスタ(IFR)

IFRは、読み出しと、ビットクリアのできるレジスタです。チップセレクトされてこのレジスタが指定されると、IFRの内容はデータバスに乗ります。ビット7はIRQの状態を示し、次のような論理式で表わされます。

IRQ=IFR6×IER6+IFR5×IER5+IFR4×IER4+·····+1FRØ×IERØ

 $(\times \text{tAND}, +\text{tOR})$  ビット6から $\emptyset$ まではラッチになっていて、次の条件でセット/クリアされます。

IFRのビット7はフラグではないので、ここに1を書き込んでもクリアすることはできません。このビットをクリアするには、レジスタの他のビットをクリアするか、次の章で説明するように、起っている割込みをすべてディスエーブルするかしなければなりません。

| 21 | セット条件                   | クリア条件                         |
|----|-------------------------|-------------------------------|
| Ø  | CA2ピンの信号のアクティブトランジション   | Aボート出力レジスタ (ORA) への読み出し又は書き込み |
| 1  | CAI ピンの信号のアクティブ・トランジション | Aボート出力レジスタ (ORA) への読み出し又は書き込み |
| 2  | シフト8回終了                 | シフトレジスタへの読み出し又は響き込み           |
| 3  | C82ビンの信号のアクティブ・トランジション  | 8ボート出力レジスタ (ORB) への続み出し又は響き込み |
| 4  | CB1ピンの信号のアクティブ・トランシショラ  | Bボート出力レシスタ (ORB) への鉄み出し又は實き込み |
| 5  | タイマ2のタイマアウト             | T2下位カウンタ鋏み出し<br>T2上位カウンタ書き込み  |
| 6  | †<br>  タイマ1のタイムアウト      | T1下位カウンタ読み出し<br>T1上位カウンタ書き込み  |

# H-2. 割込みイネーブルレジスタ(IER)

制込みイネーブルレジスタの各ビットは、IFRの各制込みフラグに対応しています。プロセッサはアドレス1110(IERアドレス)への書き込みで、このレジスタのビットを個別にセットしたリクリアしたりすることによって、他に影響を与えずに個別に割込みをコントロールすることができます。

IERへ書き込むデータのビット7がりならば、他の1のビットの割込みイネーブルはクリアされ、りのビットについては変化を生じません。逆にIERへ書き込むデータのビット7が1ならば、他の1のビットの割込みイネーブルはセットされ、りのビットについては変化を生じません。このビットを個別にセット、クリアできる機能はシステムの割込みコントロールに大変便利です。

プロセッサはこの他にIERの内容を読み出すこともできます。この時ビット7は Ø となります。

# 機能コントロール

### 1.機能コントロール

MPS6522の機能と動作モードは、ペリフェラルコントロールレジスタ (PCR) と補助コントロールレジスタ (ACR) の「つのレジスタが主にコントロールします。PCRは4本のペリフェラルコントロールピンの動作を、ACRは二つのインターバルタイマ (T1、T2) と直列ポートの動作モードを主にコントロールします。

### I-1. ペリフェラルコントロールレジスタ

ペリフェラルコントロールレジスタは次のような構成になっています。

| Eok | 7 | Б         | 5 | 4             | 3 | 2         | 1 | Ø             |
|-----|---|-----------|---|---------------|---|-----------|---|---------------|
| 機能  |   | 382⇒>トロール | V | Ċ81<br>コントロール | 6 | CA2コントロール |   | CAI<br>コントロール |

これらの機能を詳しく説明します。

### 1. CA1コントロール

PCRのビット Ø は、CA1の割込み入力信号のアクティブトランジションの方向を決めます。もしこのビットが Ø ならCA1の割込みフラグは入力パルスの立ち下がり、(Hから L)でセットされ、1 なら立ち上がり(Lから H)でセットされます。

### 2. CA2コントロール

CA2ピンは割込み入力としてもペリフェラルコントロールとしても使えます。 入力モードには割込みフラグのリセットの方法によって異なる2つのモードがあ り、どちらもCA1と同様アクティブトランジションの方向を切替ることができま す。

出力モードではCA2をCB2と組合わせて働かせることができます。これによってプロセッサはCB1とCB2が先に説明したシリアル動作をするシステムで通常の

| PCR3 | PCR2 | PCR1 | <del>Ε</del> - κ  |
|------|------|------|---|
| 0    | pl   | β    | 入力モード CA2前込みフラグ(IFR®)は入力パルスの立ち下がりでセット<br>され、ORAへの読み出し又は書き込みでクリアされる。               |
| 9    | gi   | 1    | 独立制込み入力モード IFRØはCAZの入力パルスの立ち下がりでセットされるが、ORAへ読み出し、書き込みを行なってもクリアされない。               |
| ø    | 1    | 8    | 入力モード HROはCA2の入力パルスの立ち上がりでセットされ、GRAへ腰み出し、食き込みを行なえばクリアされる。                         |
| В    | 1    | 1    | 独立制込み入力モード IFRのはCA2の入力パルスの立ち上がりでセットされるが、ORAへ読み出し、書き込みを行なってもクリアされない。               |
| 1    | ø    | g    | ハンドシェイク出力モード CA2はORAへの読み出し又は書き込みによって"L"にセットされ、CA1に入力パルスが来るとそのアクティブトランジシマンで"H"に戻る。 |
| 1    | ø    | 1    | バルス出力モード。ORAの読み出し、書き込みを行なうと、次の1サイク<br>ルだけCA2は"C"になる。                              |
| 1    | 1    | ø    | マニュアル出力モード。CA2を"L"にする。  |
| 1    | 1    | 1    | マニュアル出力モード CA2を"H"にする。  |

書き込みハンドシェイクをさせることができるのです。CA2の動作モードを上に示します。

独立割込み入力モードでは、ORAレジスタへ読み出し、書き込みをおこなっても、CA2割込みフラグはクリアされません。このフラグをクリアするには、IFRの当該ビットに1を書き込んでやらねばなりません。このモードによってプロセッサは、ペリフェラルI/Oポートでおこなわれている動作からは独立して割込み信号を受付けることができます。

ハンドシェイクとパルス出力モードについては前に説明した通りです。出力信 号のタイミングは、動作が読み出しか書き込みかによって少し違います。

### 3. CB1コントロール

CB1のアクティブ、トランジションのコントロールはCA1の場合とまったく同じです。CB1割込みフラグ (IFR4) は、PCR4が例ならば、CB1入力の立ち下がりで、1ならば立ち上がりでセットされ、どちらの場合もORBレジスタの読み出しまたは書き込みでクリアされます。

シフトレジスタがイネーブルされていると、CB1はシフトレジスタのクロックの人出力ピンとなりますが、この時もIFR4はCB1の信号のどちらかのトランジションに対して応答します。

### 4. CB2コントロール

シリアルポートがディスエーブルの時、CB2ピンの動作はPCRの上位3ビット

が決定します。CB2のモードはCA2の場合とほとんど同じです。

| PCR7 | PCR6 | PCR5 | ±− k.  |
|------|------|------|--|
| g    | ø    | р    | 新込み入力モード CB2の割込みフラグ年FR3)はCB2入力の立ち下がりでセットされ、ペリフェラルBの出力レジスタ(QRB)への読み出し又は響き込みでクリアされる。 |
| 0    | Ø    | 1    | 単独劇込み入力モード I FR3はCB2入力パルスの立ち下がりでセットされる<br>が、QRBへの挟み出し又は書き込みではクリアされない。              |
| 0    | 1    | ø    | 朝込み入力モード IFR3はCB2入力パルスの立ち上がりでセットされ、<br>QRBへの腰み出し又は響き込みでクリアされる。                     |
| 0    | 1 .  | 1    | 単強割込み入力モード 1FR3はCB2入力パルスの立ち上がりでセットされるが、ORBへの読み出し、書き込みではクリアされない。                    |
| 1    | ø    | 9    | ハンドシェイク出力モード CB2は、ORBへの書き込みで"じ"になり、<br>CB1入力パルスのアクティブ・トランジシマンで"H"に戻る。              |
| 1    | g    | 1    | パルス出力モード ORBへの着き込み後の1サイクルだけCB2が"L"になる。   |
| 1    | 1    | 9    | マニュアル出力モード CB2 € "L" にする。  |
| 1    | 1    | 1    | マニュアル出力モード CB2を "H" にする。   |

# I-2. 補助コントセールレジスタ

補助コントロールレジスタの機能についてはすでにほとんど説明してありますがここに改めて一括して説明します。このレジスタの構成は次の通りです。

| ピット | 7     | 6       | 5            | 4 | 3               |    | 2 | 1     | · · ·          |
|-----|-------|---------|--------------|---|-----------------|----|---|-------|----------------|
| 機能  | ועבוז | - □ - ル | f2<br>コントロール |   | シフトレジ:<br>コントロー | スタ |   | PBラッチ | PAラッチ<br>イネーブル |

### 1 PAラッチイネーブル

MPS6522では、PAポートもPBポートも入力をラッチすることができます。このモードでは、PAポートの入力ピンのデータはCA1の割込みフラグがセットされた時点でラッチされ、PAポート読み出し時にプロセッサへ転送されます。CA1の割込みフラグがセットされている状態でPAポートの人力データが変化してもラッチの内容は変りません。この入力ラッチはCA2が入力モードでも出力モードでも使えます。

PAポートではプロセッサはペリフェラルピンのデータを読む時は常にラッチを介します。このためPAポートが出力モードの時は読み直す時にORAの内容がラッチに正しく入っているとは限りません。このようにPAポートで出力モードと入力ラッチを組合せて使う時はシステム動作の設計上注意が必要です。

# 2. PBラッチイネーブル

PBポートの入力ラッチも、PAポートの場合と同様にコントロールされます。 ただしPBポートでは、ラッチの同容がペリフェラルピンの電圧になるかORB の内容になるかはそのピンが入力モードか出力モードかによって違います。そしてPAポートの場合と同様、プロセッサは常にラッチの内容を読みます。

# 3. シフトレジスタコントロール

シフトレジスタの動作モードは次のように決定されます。

| ACR4 | ACR3 | ACR2 : | €-k                |
|------|------|--------|--------------------|
| В    | 0    | 0      | シフトレジスタディスエーブル     |
| Ø    | 0    | 1      | タイマ2によるシフト入力       |
| 8    | 1    | Ø      | システムクロックによるシフト入力   |
| Ø    | 1 .  | 1      | 外部クロックによるシフト入力     |
| 1    | ß    | ø      | タイマ2による ブリーランニング出力 |
| 1    | Ø    | 1      | タイマ2によるシフト出方       |
| 1    | 1    | р      | システムクロックによるシフト出力   |
| 1    | 1    | 1      | 外部グロックによるシフト出力     |

# 4. T2コントロール

タイマ2の動作には2つのモードがあります。ACR5が $\emptyset$ だとワンショットモードのインターバルタイマになり、1だとPB6の入力パルスをあらかじめ与えられた数だけカウントします。

### 5 T1コントロール

タイマ1の動作モードは、ワンショットかフリーランニングかの切替えと、 PB7の出力をイネーブルにするかディスエーブルにするかの切替えで下の表のように決まります。

| <br>ACR7<br>因力イネーブル | スリーランニング<br>イネーブル | ₹→K                                       |
|---------------------|-------------------|---|
| B                   | 0                 | T1がロードされる毎にタイムアウトで単発の割込み発生<br>PB7はディスエーブル |
| Ø                   | 1                 | 連終制込み発生。PB7はディスエーブル                       |
| 1                   | 9                 | T1がロードされる毎に 単発制込みを発生し、<br>PB7にパルスを出力      |
| 1                   | 1                 | 連続制込みを発生し、PB7に方形液出力                       |

# MPS6522の応用

MPS6522はマイクロプロセッサ用汎用I/Oとして大変優れていますが、その強力な機能は複雑な動作モードと結びついているため、簡単な説明だけでは理解が困難です。ここでは、このチップがマイクロプロセッサシステムの中でどのように使われるかを示して、システム設計の理解の助けとします。

### A. MPS6522の割込みコントロール

MPS6522では割込みサービスを容易におこなえるよう、すべての割込みフラグは一つのレジスタにまとめられています。チップ内の7つの割込み発生源に対し、IRQ出力は1本なので、プロセッサは割込みを起した発生源を知るためにフラグを調べなければなりません。このためフラグレジスタの内容をアキュムレータに読み込むのですが、その時場合によっては割込みイネーブルレジスタでディスエーブルされているフラグをマスクして除去しておく必要があります。これは特に割込みがイネーブルでもディスエーブルでもフラグがセットされてしまうエッジ検出の人力動作では重要です。フラグをマスクするには、アキュムレータとIERとのANDを取るか、ANDイミディエート命令を実行します。

この結果フラグがセットされていれば、チップ内で実際に割込みが起っています。この割込みを検出するにはシフトとブランチ命令を続けて実行します。

割込みフラグをクリアするのは簡単で、割込みレジスタの当該ビットに1を書けばよいのです。これは割込みイネーブル / ディスエーブル操作といっしょに次のようにおこなえます。

LDA #@1000100000 : アキュムレータイニシャライズ

STAIFR : 割込みフラグクリア

STAIER : 割込みイネーブルフラグセット

または、

LDA #@00001000 : アキュムレータイニシャライズ

STAIFR : 割込みフラグクリア STAIER : 割込みディスエーブル

もう一つの方法は、読み込んだフラグレジスタの内容をそのまま送り返してやる ものです。

LDA IFR : IFRからアキュムレータへ転送 STA [ER : 発生した制込みのフラグをクリア

この操作の後もアキュムレータには測込みフラグの情報は残ります。そしてフラグレジスタへの書き込みではすでにセットされているフラグだけがクリアされますので、セットされつつあるフラグをクリアしてしまうという誤動作の可能性が少なくなります。

# B. タイマ1使用例

タイマ1もMPS6522の強力な機能の一つです。一定間隔での割込み動作とPB 7の電圧のコントロール機能によって、いろいろなタイミング動作、データ検出、波形発生などをおこなわせることができます。

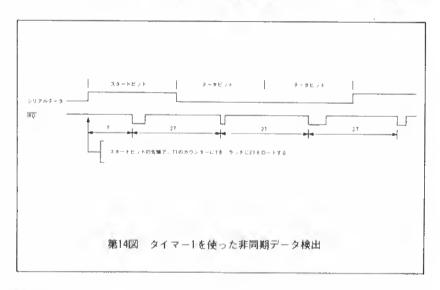
# B-1. 時刻表示時計

時刻を表わす時計の機能が必要な場合がよくあります。マイクロプロセッサシステムでは、この機能は通常メモリの中に置かれて、プロセッサに定期的に割込みをかけて割込みサービスルーチンで校正し、メインプログラムで必要な時はいつでも時刻が得られるようにします。

これまでのタイマを使った定期的な割込み動作では、割込みのたびにタイマをロードしなければなりませんでした。さらに、割込みへの応答時間が一定しないと割込みの周期も変動しました。このような問題はタイマ1のフリーランニングモードの割込みではすべて解決されて、クロック周波数の変動以外の時計の誤差の原因はなくなりました。

# B-2、非同期データの検出

非同期シリアルのアスキー信号や、データ収録機器からのシリアルデータ、クロックの検出では、正確にストローブパルスを発生することが重要です。従来のタイマでは前に述べた通り割込み応答時間によってストローブの周期が変動しますが、T1ならば正確な割込みを起すことができ、プロセッサはこの割込みに対して入力データをストローブします。この場合、進行中のカウントダウン動作に



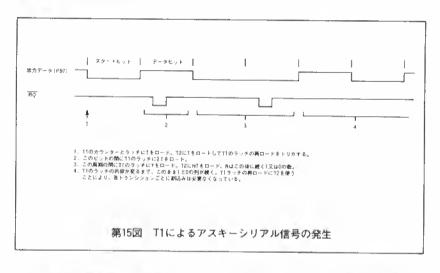
関係なくT1のラッチをロードしなおすことができるのは大変有利で、データ検出の動作中に次のストローブ時間を倍にしたり半分にしたりすることができます。この動作のタイミング図を第14図に示します。

# B-3. タイム1による波形発生

T1はプロセッサに割込みをかけるだけでなく、ペリフェラルピンPB7の出力電圧をコントロールすることもできます。ワンショットモードでは単発のネガティブパルスを、フリーランニングモードでは連続波形を発生します。フリーランニングの時はT1のタイムアウトのたびにPB7のレベルが反転します。

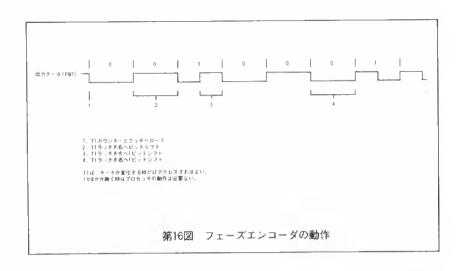
ワンショットモードで、PB7の出力によってソレノイドを直接トリがすることができます。T1C-Hへの書き込みのたびにソレノイドはトリガされます。

ラリーランニングモードでカウントダウン中にラッチにデータをロードして次のカウントダウンの長さを決定することにより、PB7に複雑な波形を発生させることもできます。この方法でアスキーのシリアルデータを発生させるタイミング 図を第15図に示します。



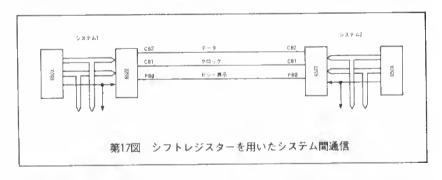
この方法は磁気テープやディスクのフェーズ・エンコーダにも応用できます。 フェーズ・エンコーダの動作のタイミング図を第16図に示します。

ここに示した応用例はT1の能力から見ればほんの一部に過ぎません。他にも、 パルス幅変調、ビデオゲームの効果音の発生、正確なパルス幅の必要なA/D枝 術、電子ゲームの波形発生などへの応用が可能です。

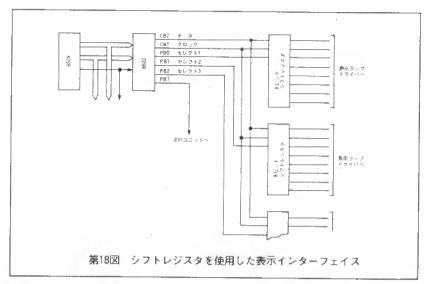


# C.シフトレジスタの応用

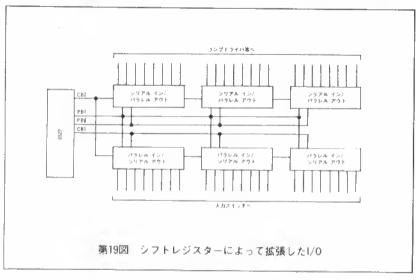
MPS6522のシフトレジスタは、主にシステム間の同期式シリアルデータ通信用に設計されています。通信をおこなうシステムはシングル・プロセッサで複数のペリフェラル・コントローラを持つシステムの場合もありますしマルチプロセッサ・システムの場合もあります。このシフトレジスタは、CRによってノイズを抑えて比較的低速でのデータ転送に適しており、データ転送はプロセッサが他の仕事をしている間におこなうことができます。2つのプロセッサを持つシステムの例を第17図に示します。MPS6522のシフトレジスタにより、複雑な非周期通信技術を使わないでシステム間の通信をおこなっています。



複数の周辺機器を持つシステムでは、シフトレジスタを使ってこれらのインターフェイスにデータを送ることができます。第18以に、複数のステータス表示を



持ったシステムの例を示します。このシステムではそれぞれ独立したコントローラが簡単なドライバを介してステータス表示のランプを働かせます。データ線とクロック線は各ユニットに並列に接続され、さらにポートBの出力でデータを入れるユニットを選択していますが、どのユニットも同じ表示をするならこの選択線は必要ありません。図に示したシステムでは、ユニットを指定し、シフトレジスタへデータを書き込むことによって表示を新しくするようになっています。



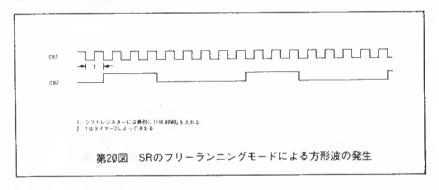
入力機器の制御もまったく同様で、第18図のようにペリフェラルポートの出力のコントロールにより、データをシフトレジスタへ入力することができます。まず読みたい入力のユニットを選択し、シフトレジスタをアクセスするとシフト動作が始まり、終了すると割込みがかかってプロセッサはデータを読み取ります。

上に述べた方法はシステムのI/O機能を拡大するのに役立ちます。ステータス表示のランプと入力データスイッチを多く備えたシステムでも、簡単なTTLのシフトレジスタを使って低価格で構成することができます。この例を第19以に示します。

# シフトレジスタを使った方形波の発生

出力モードでは出力されるデータのビット7は再びビット Ø に戻りますので、 同じデータを連続して出力する時はシフトレジスタをロードしなおす必要はあり ません。シフト動作はシフトレジスタを読み出すことによって始まります。

この機能を使うと8ビットのパターンを続けて出力することができるので、周辺機器のクロックとして使えます。この方法を第20図に示します。この時シフト動作はタイマ2によってコントロールされるので、1ビットの時間は最大でクロックパルスの256倍となります。



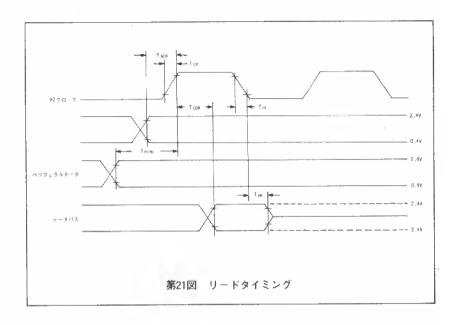
# 最大定格

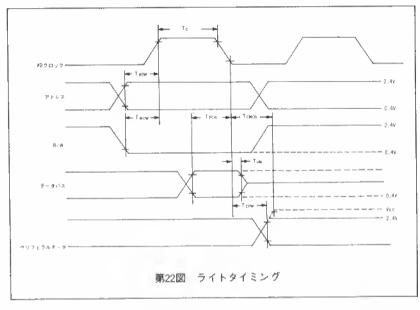
| 項    | B | 尼号   | 定格        |     |
|------|---|------|-----------|-----|
| 電源電圧 |   | Yee  |           |     |
| 入力電圧 |   |      | -0.3-+7.0 | Vdc |
| 動作温度 |   | TA   | 0-70      | ·c  |
| 保存温度 |   | Taig | -55~+150  | 'C  |

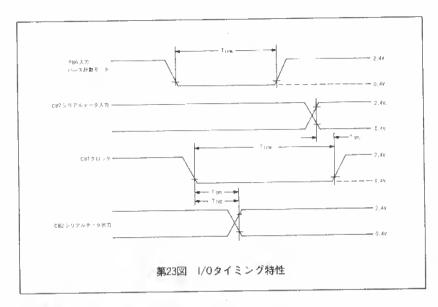
このチップには静電気に対する保護回路がついていますが、最大定格を越える電圧をかけないよう注意して下さい。

直流特性 (特に記載のない場合はVcc=5.0V±5%、Uss=0、T<sub>A</sub>=0~+70℃)

| 項 目  | 12 =            | 施 少  | 存 準    | 8 *  | * 17   |
|--|-----------------|------|--------|------|--------|
| ·H. 入力電圧                                       | Viн             | +2.4 |        | Yee  | Vdu    |
| · H. 入力電圧                                      | V <sub>II</sub> | -0.3 | -      | 10.4 | Vds    |
| 入力リーク電池 Yin=ポー5Vdc                             | Eph             |      | ±1.0   | ±2.5 | µAdc   |
| R/W. RES, RSØ, RS1, RS2, RS3, CS1 CS2, CA1, Ø2 |                 | ~    | ±2.0   | ± 10 |        |
| オフ状態入力電流 Yxx=0.4~-2.4V                         | Irsi            | -    | +2.0   | ±10  | µAdc.  |
| Vcc≃Max, D♦ D7                                 |                 |      |        |      |        |
| (H. 人力看班 V <sub>H</sub> =2_4V                  | (p)             | -100 | -250   | -    | prAdc. |
| PAG-PA7, CA2, PBG-PB7, CB1, CB2                |                 |      |        |      |        |
| L. 人力電池 V <sub>IL</sub> =0.4V                  | l <sub>IL</sub> | _    | ~1.0   | -1.6 | mAdc   |
| PAB PA7, CA2, PB6-PB7, CB1, CB2                |                 |      |        |      |        |
| H 出力電社 Vcc=最小 Hoad=-ICOuA                      | V <sub>OH</sub> | 2.4  | _      |      | Vdc    |
| PA@-PA7,CA2,P88 P87,C81,C82                    |                 |      |        |      |        |
| 1.出力電圧 Vcc=最小 Noad=1.6mAdc                     | Ψα              | -    | -      | 10.4 | Yac    |
| H-出力権連(ワース)                                    | LOH             |      |        |      |        |
| V <sub>DH</sub> =2.4V                          |                 | -100 | - 1000 | -    | μAdc   |
| V <sub>OH</sub> — 1,5¥, PBØ- PB7, CB1, CB2     |                 | -3.0 | - 5.0  | -    | mAdc   |
| し、出力電池(シンク) V <sub>OL</sub> =0.4V              | lor             | 1.6  | -      | _    | mAdc   |
| オフ秋歌 出力リーク電波 (RQ)                              | loff            |      | 1.0    | 10   | µAdc   |
| 入力容量 'Ta=25°C f-1MHz                           | Cm              |      |        | _    |        |
| R/W. RES. RSB, RS1, RS2, RS3, CS1, CS2         |                 | _    | -      | 7.0  | PF     |
| D#-07, PA# PA7, CA2, P8# P87, C81, C82         |                 | -    | -      | 10   | PF     |
| 92   |                 |      | -      | 20   | Pf     |
| 出力容量 T <sub>A</sub> =25°C (= IMH <sub>E</sub>  | Coul            |      | ]      | 10   | PF     |
| 消費電力   | Po              |      |        | 1000 | ΨW     |







# 交流特性

# リードタイミング (第21図、負荷130pFおよび1 TTLロード)

| 項目                          | 起    | 뮴 | <del>55</del> | 小 | 標 | ?# | ñ   | * | 車    | 垃 |
|-----------------------------|------|---|---------------|---|---|----|-----|---|------|---|
| アトレス確定からクロック立ち上がりまでの遅延時間    | TACR |   | 180           |   | - |    | -   |   | ns   |   |
| クロック立ち上がりからバスにデータが乗るまでの選延時間 | Toda |   | -             |   |   |    | 395 |   | ns   |   |
| ペリフェラルデータのセットアップ時間          | TPCR |   | 300           |   | - |    | _   |   | P.S. |   |
| データバス保持時間                   | THR  |   | 10            |   | _ |    | _   |   | ns   |   |
| クロック入力立ち上がり、立ち下がり時間         | Ton  |   |               |   | - |    | 25  |   | ns.  |   |

# ライトタイミング(第22図)

| 頃 目  | 12 号  | ■ 小  | 標準 | 易大  | 单位  |
|--|-------|------|----|-----|-----|
| イネーブルバルス市                                    | Тс    | 0.47 |    | 25  | μs  |
| アドレス確定からクロック立ち上かりまでの遅延時間                     | Tacw  | 160  |    | _   | ns  |
| テータ確定からプロック立ち下がりまでの遅延時間                      | Toca  | 300  | -  |     | пs  |
| B/W立ち下がりからグロック立ち上がりまでの遅延時間                   | Twew  | 180  | -  | -   | nş  |
| テータバス保持時間                                    | THW   | 10   |    |     | ns. |
| イネーフル立ち下かりからペリフィラルテータ確定までの<br>液晶時間           | TOPM  |      |    | 1.0 | # h |
| CMUS(Vcc30%)」、おけるクロック立ち下かりからテータ<br>確定すでの遅延時間 | Temos |      |    | 2.0 | μ1  |

# ペリフェラルインターフェース特性

| 頃  | E 55             | 表 小 | 標準 | 最大   | 单 位  |
|--|------------------|-----|----|------|------|
| CA1, CB1, CA2, CB2 入力の<br>立ち上がり、立ち下がり時間                  | TRF              | -   | -  | 1.0  | . µs |
| クロック立ち下かりからCA2立ち下かりまでの<br>運延特間(読み出しハンドシェイク又はパルスモード)      | TCAZ             | -   |    | 1.0  | μs   |
| クロック立ち下かりからCA2立ち上かりまでの<br>遷延時間(パルスモード)                   | Test             |     | -  | 1,0  | μ5   |
| GATアクティブ・トランジションからCA2立ち上がり<br>までの遅延時間 (ハンドシェイクモード)       | T RS2            |     | -  | 2.0  | μς   |
| クロック立ち上がりからCA2又はCB2立ち下がり<br>までの遅延時間(書き込みハンドシェイク)         | T WHS            | _   | -  | 1.0  | μς   |
| ベリフェラルデータ確定からCB2立ち下がりまでの<br>漢廷時間                         | Toc              | 0   | _  | 1,5  | μs   |
| クロック立ち上がりからCA2, CB2立ち上がりまでの<br>遅延時間 (パルスモード)             | T <sub>PS3</sub> | _   |    | 1.0  | μ5   |
| CB1アクティブ・トランジションからCA2,CB2<br>立ち上がりまでの遅延時間 (ハンドシェイクモード)   | T RS4            | -   | _  | 2.0  | μs   |
| ペリフェラルデータ確定からCA1,CB1アクティブ・<br>トランジションまでの遅延時間 (入力ラッチ)     | T <sub>IL</sub>  | 300 | _  | -    | ns   |
| CB1立ち下がりからCB2データ確定からCB2データ<br>確定までの遅延時間 (内部SRクロック、シフト出力) | TsRI             | _   |    | 30C  | ns   |
| CB1入力クロック立ち下がりからCB2データ確定まで<br>の遅延時間(外部クロック・シフト出力)        | T <sub>SR2</sub> | -   | _  | 300  | ns   |
| C82データ確定からCB1クロック立ち上がりまでの<br>遅延時間 (外部又は内部クロックのシフト入力)     | Tspa             | -   | -  | 300  | ns   |
| P86入力パルスのパルス巾  | Tipw             | 2   | -  | _    | μ5   |
| CBI入力クロックのパルス市   | Ticw             | 2   | -  | 34.6 | иѕ   |
| F86入力パルスの間隔  | Ties             | 2   | _  | -    | μs   |
| CBI 入力パルスの間隔   | Ties             | 2   | _  | _    | из   |

# MPS 2364 スタティックROM(8192×8)

# 概要

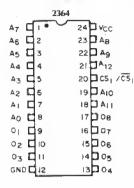
MPS 2364は、65,536ビットのスタティック、リード・オンリーメモリーで、8192×8ビットで構成されています。アクセスタイムが、短い (最大35gns)という特徴をもちます。ハイ・パーフォーマンス、大容量、簡単なインタフェイスが重要視される応用製品、およびマイクロプロセッサーとコンパチブルであるように設計されています。

2364は、非同期に動作し、クロック入力を必要としません。プログラマブル・チップ・セレクト入力1つにより、2つの64KROMが外部のデコードに回路なしにORで結ばれます。2個の2732EPROMとおきかえることにより、いったんEPROMでプロトタイプされたプリント板をマスク・プログラムROMにかえることによる再設計の必要がありません。

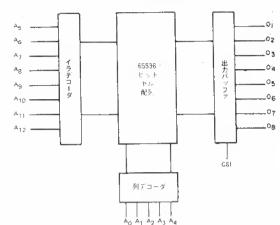
# 特長

- ○アクセスタイムは45@nsおよび35@ns
- ○スタティック動作
- ○TTLコンパチブル
- ○ワイヤーOR増設に対してスリーステート出力
- ○プログラマブル・チップ・セレスト
- ○5 V単一電源
- ○2716および2732EPROMとピンコンパチブル
- ○400mV入力ノイズを無視
- ○2716 / 2732EPROM をプログラムデータ入力として受入可能

# ピン配置図



# ブロック図



### 直流特性 (特に記載のない場合は、T<sub>A</sub> = 8 °C ~ +78 °C 、V<sub>CC</sub> = 5.8 V ± 5%)

| 5 号              | 項目       | 最小   | 最大                 | 単位   | テスト条件   |
|------------------|----------|------|--------------------|------|---|
| Iges             | 供給電流     |      | 125                | m A  | V <sub>N</sub> =V <sub>CC</sub> 、V <sub>0</sub> =オープン、T <sub>A</sub> =Ø℃    |
| Icea             | 供給電流     |      | 129                | mΑ   | $V_{NN} = V_{CC}$ , $V_{CI} = A - \mathcal{I} \mathcal{I}$ , $T_{A} = 25\%$ |
| l <sub>0</sub>   | 出カリーク電流  | í    | FB                 | μA   | チップ・ディセレクト、Vo=ØからVoc  |
| J <sub>i</sub> i | 入力ロード電流  |      | 10                 | μA   | V <sub>CC</sub> =最大、V <sub>IN</sub> =ほからV <sub>CC</sub>                     |
| Vo.              | ·L"出力電圧  |      | 9.4                | Volt | V <sub>GC</sub> = 表小、1 <sub>0L</sub> = 2.1mA                                |
| V <sub>OH</sub>  | 'H''出力電圧 | 2.4  |                    | Volt | V <sub>GG</sub> = 最小、I <sub>OH</sub> = −400 μA                              |
| VIL              | "し"入力電圧  | -Ø.5 | F.8                | Volt | 注1拳腕  |
| VIH              | *H"入力電圧  | 2.0  | V <sub>00</sub> +1 | Volt |   |

### **交流特性 (特に記載のない場合は、T₄ = 8℃ ~+78℃、Vcc =5.8V + 5%)**

|                 |            |           | 2364 |     | 2364A |     | Jil. 17 | テスト条件       |
|-----------------|------------|-----------|------|-----|-------|-----|---------|-------------|
| 記 号             | <b>将</b>   |           | 最 点  | 胜 大 |       |     | # 12    | 7 A F 3K IT |
| face            | アドレス・アクセス時 | M         |      | 450 |       | 35Ø | nS      |             |
| ten             | チップ・セレクト遅ら | 時間        |      | 200 |       | 200 | n5      | 注 2 參照      |
| T <sub>DF</sub> | チップ・ディセレクト | - 運延時間    | i    | 175 |       | 175 | nS      |             |
| ton.            | アドレス変化からデー | タの変化までの時間 | 40   |     | 40    |     | n\$     |             |

# キャパシタンス (TA=25℃、f=1.0MHz、注3参照)

| 記号  | 頂     | E     | 嚴 | 7]1 | 殿 | 大 | hit | 拉 | テスト条件               |
|-----|-------|-------|---|-----|---|---|-----|---|---------------------|
| CIN | 入力キャ/ | ジタンス  |   |     |   | 8 | p   | F | テストビン以外はすべてACグランド接地 |
|     | 出力キャル | ベシタンス |   |     | 1 | Ø | ψ   | F |                     |

- 注1、一0.5V以下の入力電圧は避けるべきで、チップの破壊の原因となります。
- 注2、1TTL+pF使用、入力変化時間:29viii

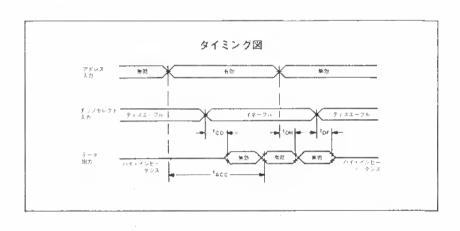
タイミング測定:入力1.5V、出力9.8Vと2.9V、CL~100pF

**川3、パラメータは周期的なサンプルであり、100%のデストではありません。** 

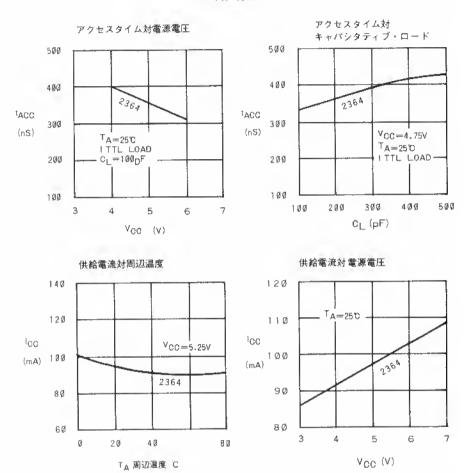
# 最大定格

| Ļ | ě | 且  | 定 格          |
|---|---|----|--------------|
| 動 | 作 | 温度 | Ø℃~70℃       |
| 保 | 存 | 温度 | -65°C -150°C |
| Œ | 源 | 電圧 | -Ø.5V ~7.ØV  |
| 出 | カ | 電圧 | -Ø.5V ~7.ØV  |
| λ | カ | 電圧 | -Ø.5V ~7.ØV  |
| 消 | 黄 | 電力 | 1.0W         |

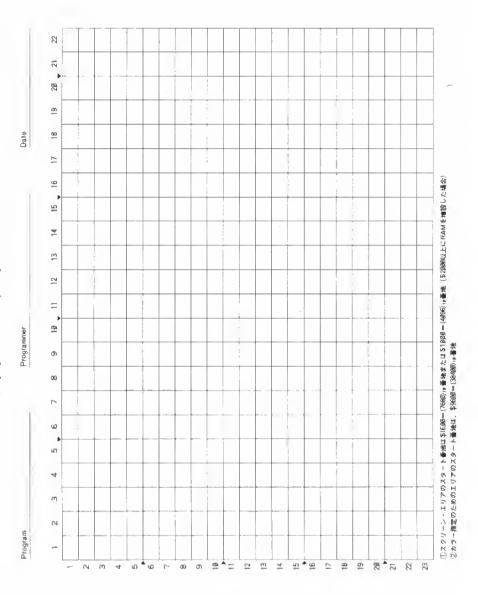
(注) 最大定格を越える場合は、チップを破壊する原因になります。これは、耐久度を述べているのであり、それまたはそれ以上の条件で、本仕様に述べている動作をするというのではありません。



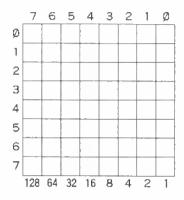
# 代表特性

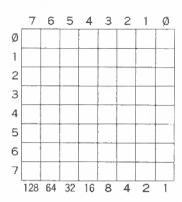


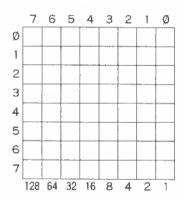
# スクリーン・レイアウト・シート

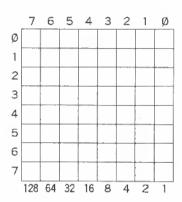


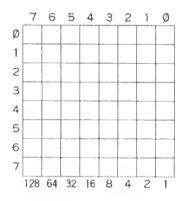
# 8×8ドット・パターン作成シート

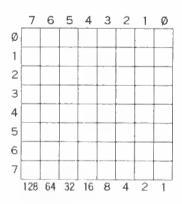




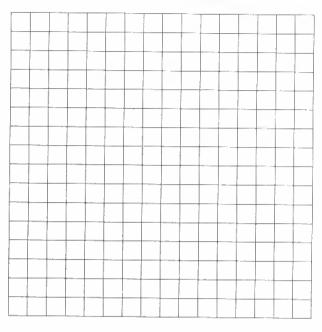


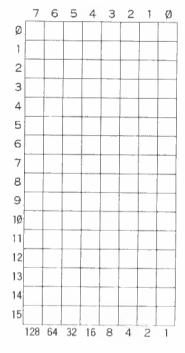


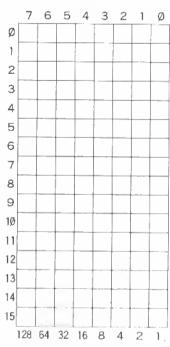




# 8×16ドット・パターン作成シート









# VIC-1001

# PERSONAL COMPUTER USER'S MANUAL

1981年1月15日第2版発行

発行所

C×コモドール・ジャパン株式会社

東京都港区赤坂8丁目5番32号赤坂山勝ビル6階 〒107 ☎03 (479) 2131

印刷所

株式会社 坂井印刷所

VIC-1001

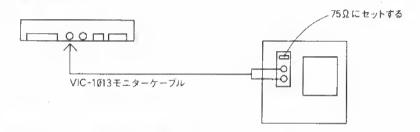
# VIC-1001とモニター・テレビ(又は家庭用カラーTV)との接続方法

まず、VIC-1001及びモニター・テレビの電源が切れている事を確認して下さい。

そして、御使用になる TVのタイプにより、下に示してある方法により、 VIC-1001と TVを接続し、TVの電源を先に入れ、その後 VIC-1001の 電源を入れて下さい。画面は約2秒以内に写し出されます。

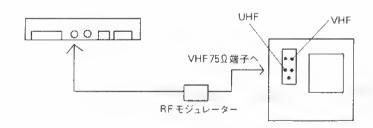
# 1. モニター・テレビ御使用の場合

VIC-1Ø13モニター・ケーブルを VIC-1ØØ1のビデオ・インターフェイス・コネクターに差し込み、一方をモニター・テレビの端子に接続して下さい。その際、モニター・テレビの端子が75Ωにセットされている事を確認して下さい。



# 2. 家庭用テレビを御使用の場合

同梱されております RFモジュレーターのコネクターを VIC-1 $\emptyset$ 01の ビデオ・インターフェイス・コネクターに差し込み、片方をテレビの VHFのアンテナ端子に接続して下さい。この際、 $75\Omega$ 端子に接続して下さい。



# テレビのカラー調整について

VIC-1001を最初に家庭用テレビまたはカラー・モニターに接続し、電源を入れたさいに、ご使用のテレビによっては、スクリーン、ボーダー、キャラクターのカラーが鮮明に出ない場合があります。このような場合には、使用テレビが家庭用テレビのさいには、カラー調整つまみ(色の濃さ、色あい、明るさ、映像など)およびチャンネル微調整つまみによって、最適のカラーになるようにご調節ください。使用テレビがカラー・モニターの場合はカラー調整つまみにより、最適のカラーになるようにご調節ください。